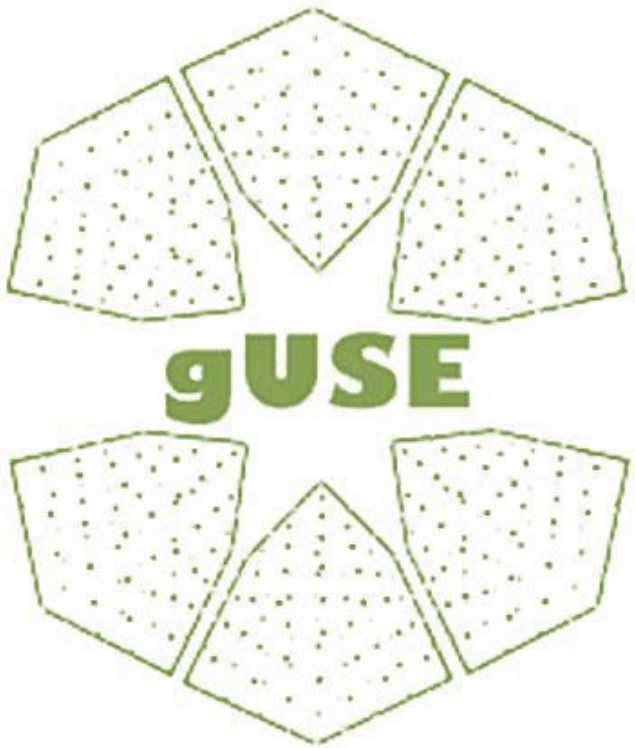
Administrator Manual and Cookbook

Guide to gUSE Components and Related Interfaces Administration

by Tibor Gottdank

Docs for Admin Series



- INSTALLATION WIZARD MANUAL
- UPGRADE MANUAL
- **ADMINISTRATOR MANUAL**
- DCI BRIDGE ADMINISTRATOR MANUAL
- REMOTE API CONFIGURATION MANUAL
- CLOUD-SPECIFIC ADMINISTRATION GUIDE
- GUSE VM IMAGE USAGE

Copyright ©2013-2014 MTA SZTAKI LPDS

MTA SZTAKI LPDS accepts no responsibility for the actions of any user. All users accept full responsibility for their usage of software products. MTA SZTAKI LPDS makes no warranty as to its use or performance.

The gUSE/WS-PGRADE is an open source software.
MTA SZTAKI LPDS inspires and supports to take the whole gUSE/WS-PGRADE community into the developing work.

Table of Contents

About this Manual 4
How to Read this Manual 4
Abbreviations 4
I. How to Start and Stop the System 5
Using of Manual Step Sequence 5
Using of Scripts 6
II. The Internal Services 7
Setting of Services 11
Setting of Service Properties 12
Quota Setting 13
Setting of DCI Bridge Service Properties 15
III. Distinction WS-PGRADE Portal from Liferay Portal 18
IV. Setup the End-User Role 20
Setting of End-User Role 20
Registering the End-User Role 21
Menu Visibility Modification 22
V. Settings on CloudBroker Platform 25
Roles 25
Registration 25
New User Creation 27
Resource Registration 29
Software Deployment 32
Wrapper Deployment 34
VI. Ticket Request to Use Data Avenue 36
VII. Embedding SHIWA Repository into WS-PGRADE 38
VIII. Additional Settings in the Case of Untrusted Certification 42
IX. Administration of Single Job Wizard 46
Single Job Wizard Setup 46

Configuration 47
Additional Enhancements 49
Adding SHIWA Repository 49
Site Page (Menu Point) Selection to WS-PGRADE in Liferay 50
The Text Editor 52
Setting System to Local Submitter 53
Robot Permission Related Logging of Job Submissions 54
About Robot Permission and Related Logs 54
Retrieving Information from Log Files 55
Adding and Removing User Roles 58
Supported Protocols for Using of Remote Executable and Input 61
New Parameter Definition to a Middleware 62
Logging with log4j 64
Administrative Tasks for Job Submission to SZTAKI Desktop Grid 65
Portlet Adding for WS-PGRADE Portal 66

About this Manual

The **Administrator Manual** acts a significant part of the gUSE/WS-PGRADE system administration information.

This guide contains the most relevant and useful information about gUSE/WS-PGRADE administration and those descriptions that are not included the other published administration documentations of gUSE/WS-PGRADE.

Finally, about a special text form: the document uses light purple color for **notifications** and light orange for **pitfalls** and **warnings**.

Note: the exact configuration details of middlewares supported by gUSE are in DCI Bridge Manual.

How to Read this Manual

The major goal of this Manual is to provide a detailed and useful administrator guide about all the main important components and interfaces of gUSE/WS-PGRADE system. This Manual is permanently further developed and improved so if you miss any important information let us know at the **SourceForge Discussion** (**http://sourceforge.net/p/guse/discussion/?source=navbar**) and we will extend the next version of the Manual with the requested information. We believe that this community-oriented editing of the Manual is the most efficient way of producing a comprehensive and user-friendly documentation.

The Liferay Portal (which is the fundamental portal technology of WS-PGRADE) administration is out of the scope of this Manual. The detailed administration documentation for Liferay Portal version 6.0 is located here: **http://www.liferay.com/documentation/liferay-portal/6.0/administration**. However, the explicitly WS-PGRADE-related Liferay settings are described in this Manual.

Abbreviations

The following abbreviations are used in the Manual:

- **EC2 –** Elastic Cloud v2
- **EGI** - European Grid Initiative
- **gUSE** - grid and cloud User Support Environment
- **OCCI –** Open Cloud Computing Interface
- **VM –** Virtual Machine

- **WFI –** Workflow Interpreter
- **WFS –** Workflow Storage
- **WS-PGRADE** - Web Service – Parallel Grid Run-time and Application Development Environment

I. How to Start and Stop the System

One of the first relevant information in an administrator manual is to explain how to start and stop the system. In case of gUSE you can do it by two ways: by manual step sequence or by starting of scripts.

Using of Manual Step Sequence

In case of manual system startup, follow the next steps:

1. Start the Apache Tomcat container (with default settings **~/guse/apache-tomcat-6.0.35/bin/startup.sh**) on (or on each of) the UNIX machine(s).
2. Start the **Service Wizard** by opening the following URL on the backend machine: **http://<URL_install_backend>:8080/information** Note that this operation needs authorization: user/password is “admin”/”admin”
   (You can change the password in tomcat-users.xml file, if it’s necessary.)
   The **Service Wizard** will be opened:
   - In the popup *Database Configuration* menu the MySQL server must be redefined. Please do not forget to replace the string IP by the URL of the MySQL server.
   - Define the SMTP server of the organization administrating the portal and the e-mail address of the person administrating the portal in the subsequent step. (optional step)
   - By selecting the appearing blue sphere containing a magnifier a control step will be played: The configuration of the WS-PGRADE/gUSE will be executed and animated by the wizard and the values of the most important URL-s are displayed.
   - As last step the page named *gUSE Service Wizard* is called and the success will be reported.

By now, your portal must be operational if you go to
http://<URL_install_frontend>:8080/liferay-portal-6.1.0/

To shut down the system

1. Stop the Apache Tomcat container (with default settings **~/guse/apache-tomcat-6.0.35/bin/shutdown.sh**) on (or on each of) the UNIX machine(s).
2. Check stopping of all java processes (**killall -9 java**) on (or on each of) the UNIX machine(s).

Certainly, you need first to stop then to start the system in case of system restart.

Using of Scripts

There is another way to start/stop the gUSE system: using of scripts. You need three different scripts to this process.

1. **guse** init script (works with Debian and with minor modifications should work in Red Hat-based distros, as well).
2. **start.sh** script to startup the system
3. **stop.sh** script to shut down the system

Using of scripts:

1. Put **start.sh** and **stop.sh** into the home directory of the user that runs the gUSE service, e.g. in **/home/guse**.
2. Put **guse** script in **/etc/init.d** and set the **GUSE_USER** variable in it to the name of the user that runs the gUSE service. Execute **/etc/init.d/guse start** as root to start gUSE and **/etc/init.d/guse stop** to stop gUSE.

Note: the **start.sh** checks if the current IP of the 'outside' interface is the same as detected during a previous startup (read from **~/.prev_init_IP**). If it's not the same, some MySQL tables containing service URLs are dropped and gUSE is initialized using the new IP for the service URLs. If it's the same, the tables are not dropped and gUSE is initialized by using earlier service URLs.

You can eliminate this behavior by echoing your machine's current IP to **~/.prev_init_IP**.

The scripts are available in **/scripts/init** directory in the gUSE package (**guse-<version number>.tgz**.) located in SourceForge.

II. The Internal Services

The **Middle tier** of the gUSE/WS-PGRADE framework (Fig. II.1) contains those high level gUSE (internal) services that enable the management, store and execution of workflows.

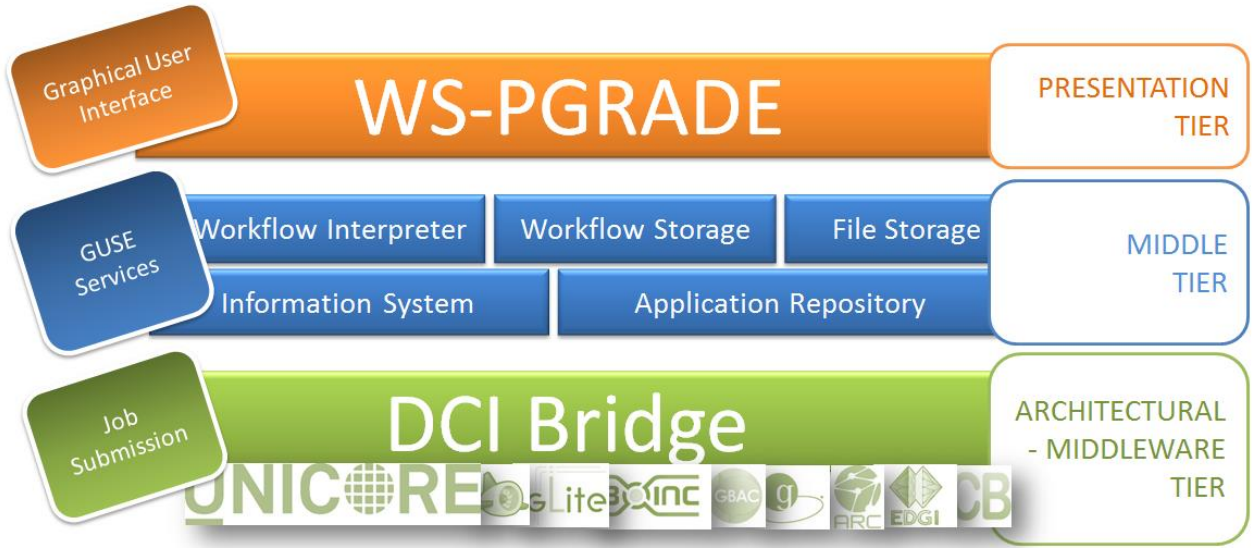


Figure II.1 The multitier architecture of gUSE/WS-PGRADE framework

The gUSE internal services are accessible only by the administrator of gUSE. The administrator can dynamically observe and tune the cooperation of internal components composing the gUSE infrastructure.

In order to configuring gUSE services, you can use as administrator the *Settings*/*Internal services* menu in WS-PGRADE. Existing service properties can be set or modified, new services can be added, connections between components can be defined, properties can be imported between existing components and the whole system configuration can be downloaded. Texts on the UI are *jstl:fmt* based with multilingual support.

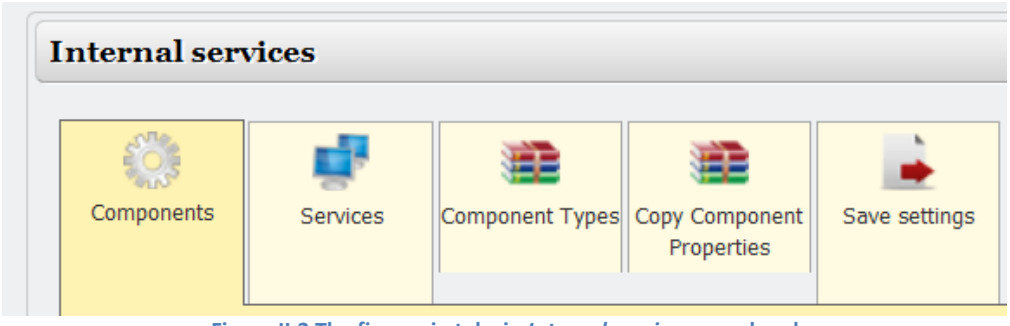


Figure II.2 The five main tabs in *Internal services* page head

The following terms have been introduced in *Internal services* function in WS-PGRADE (see the main functions as tabs on Fig. II.2):

– **Components**: Web applications of the gUSE system. Components may have many parameters which fall in two categories:
  1. Obligatory (or generic) parameters.
  2. Individual – Component Type dependent – parameters in form of key - value pairs.

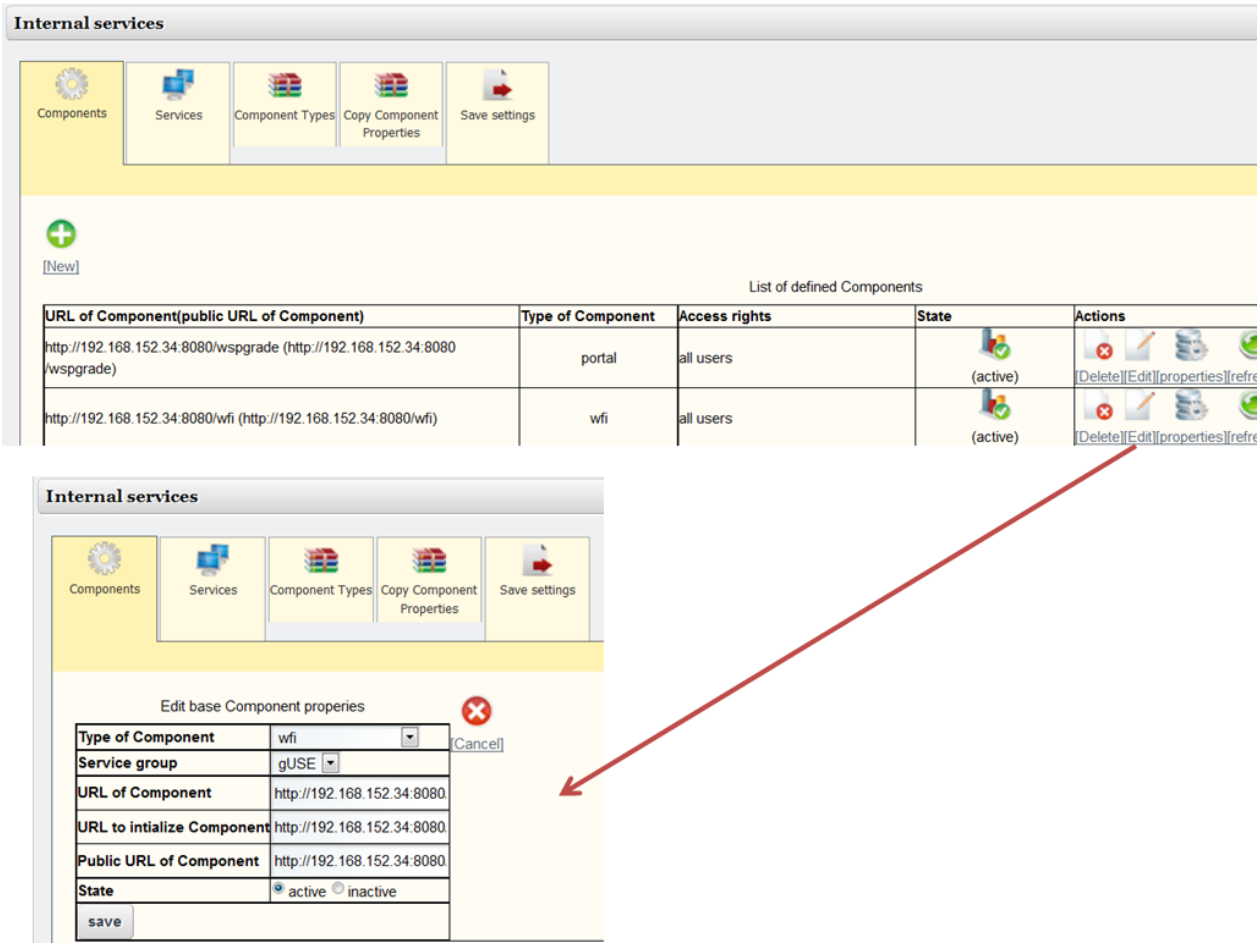


Figure II.3 Component properties

The most important obligatory parameters are (*Internal services/Components/Edit* function, lower window of Fig. II.3):

– **Type of Component**
– **Service group**: Describes the kind of protocol by which components exchange messages. At present there is just one installed protocol. Its name is "gUSE".
– **URL of Component**: Defines a hardware related access point through which internal clients of the Component's services (clients belong to the gUSE infrastructure) may access this service.
– **URL to initialize Component**: A distinguished URL to reset the Component.

- **Public URL**: Defines a hardware related access point through which external (remote) clients of the Component's services may access this service.
- **State**: Boolean variable (*Inactive/Active*)

Warning: there must be at least one Component to each base Component Type of the gUSE. It means that the needed activities must be associated to resources by which they can be performed. A Service - which must belong to a given **Service Group** - defines the possible request types among the Components.

It has four parameters:

1. Component Type of Server: It is the type of the component that serves the request of the client component.
2. Component Type of Client: It is the type of the client component that requests the service.
3. Server side service interface: It is a relative address to find the proper interface elaborating the request on the actual Component.
4. Client side interface *impl.class*: points to the definition of the Java code, which communicates with server on behalf of the client.

Further tabs in the *Internal services* window are:

- **Services**: The registry of gUSE components. (The component registration solution is similar to standard web service registration.) Here you can add or delete components in case of a gUSE system update. (Fig. II.4)
- **Component Types**: The type of web applications identified as gUSE components. (To the proper work of gUSE a predefined set of Component Types must be present.) The table of component types contains three columns: (1) The *Name of service group* is the name of group concerned the component; (2) The second column gives short information about a component type; (3) The last column provides actions (*Edit* or *Delete*) to the item. (Fig. II.5)
- **Copy Component Properties**: Enables copy the property set (identified by URL) between two selected components. (Fig. II.6)
- **Save settings**

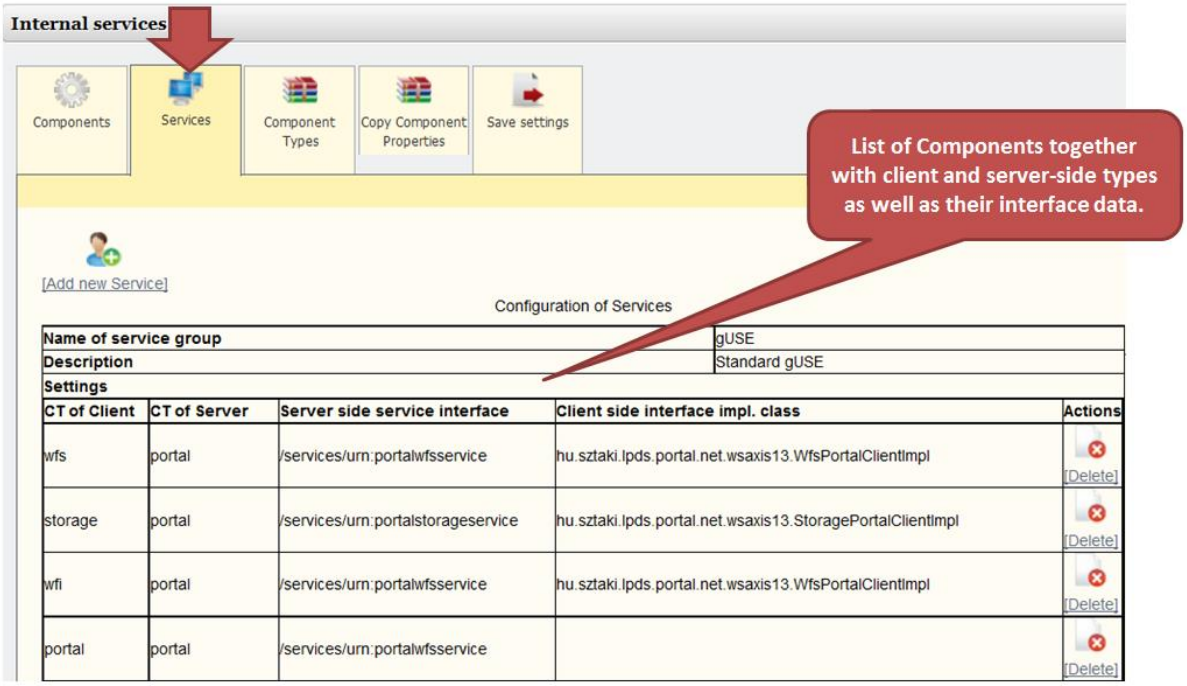


Figure II.4 The component registry in the *Services* tab

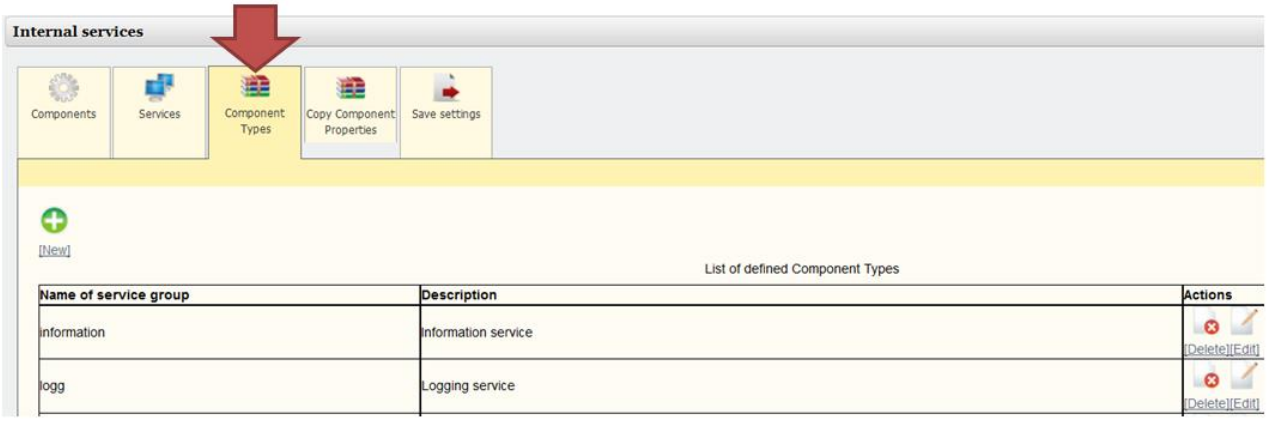


Figure II.5 The list of defined types in *Component Types* tab

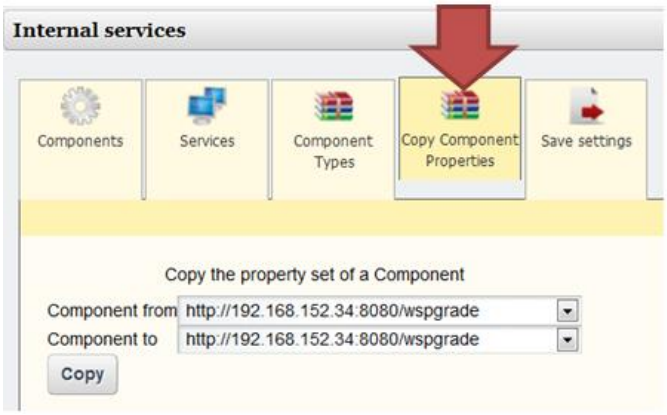


Figure II.6 The in Copy *Component Properties* tab

Setting of Services

To set services you can log as an administrator into WS-PGRADE. Click on *Settings/Internal services* (Fig. II.7).

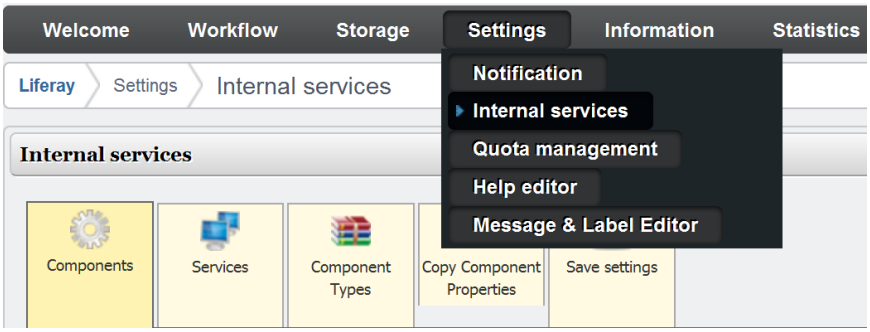


Figure II.7 Selecting *Internal services* menu

Find the service of interest (e.g., *wfi*, *wfs*), and click on *properties.* Here you can *Delete*, *Edit* or add *New* properties (Fig. II.8).

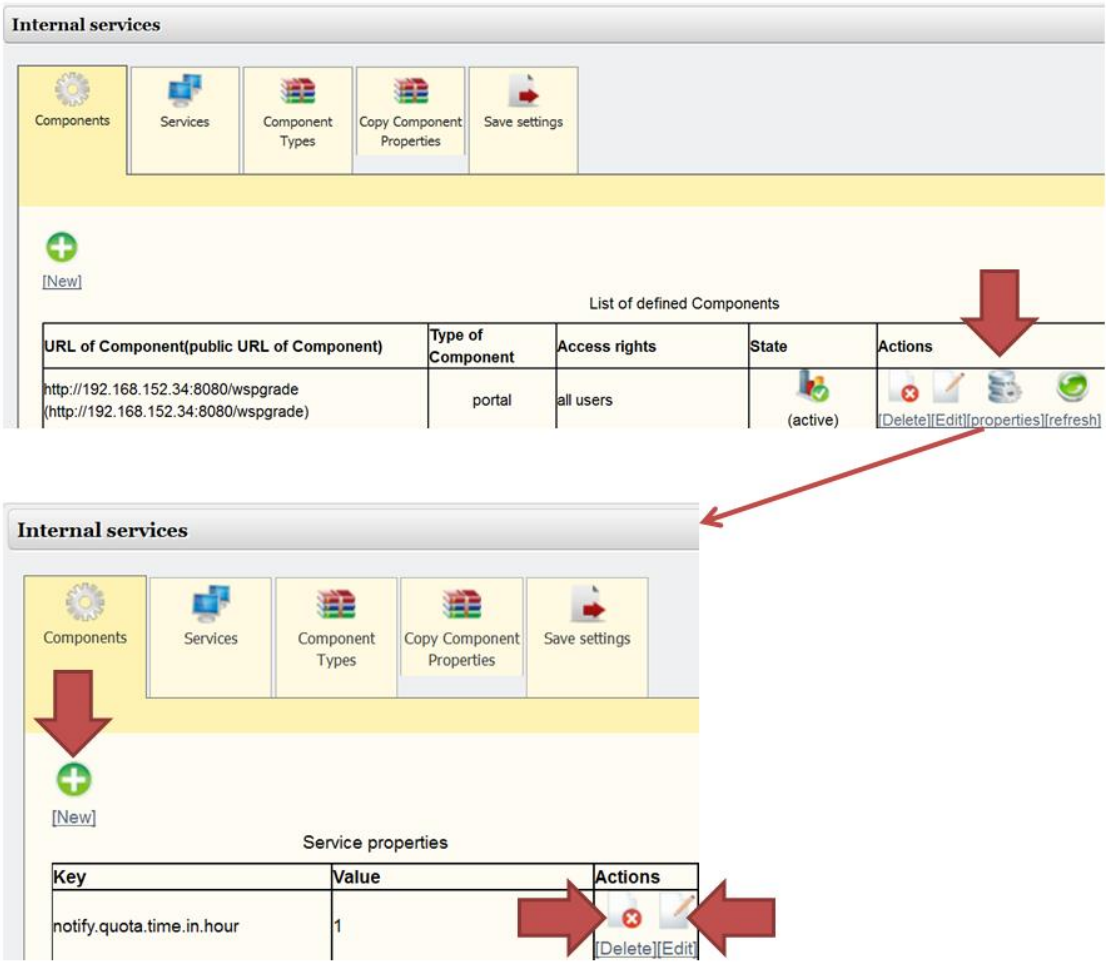


Figure II.8 The *Service properties* function within *Internal services/Components* menu

Setting of Service Properties

The most important - in respect of system performance - internal service (WFI, WFS, File Storage, DCI Bridge) properties together with their default values and descriptions are listed in the following tables (Table 1–5).

The presented WFI properties concern the used job/workflow limitation (Table 1). The WFS property (Table 2) sets the saving of the visualizer data of portal statistics. If the property value is *true*, then the job data will be saved in database server and corresponding triggers will be run. There is worth to set it to false if you don't need to use statistics because this data saving action loads the gUSE system significantly. (If you want to use the *Statistics* function of portal, you need another step: to enable the portal to show the statistics views, please navigate to page: **http://<URL install backend>:8080/stataggregator**. Thus, the collection of statistics log information is automatically switched and the *Statistics* function is usable.)

Table 1 WFI properties

Property name	Default value	Description
wfi.zen.activeingjobs.max	500000	Maximum number of jobs managed by the WFI.
wfi.zen.activeingjobs.usermax	500000	Maximum number of jobs managed by the WFI per user.
wfi.zen.quota.user.workflow	10000	Maximum number of workflows per user.

Table 2 WFS property for performance improvement

Property name	Default value	Description
guse.wfs.system.savevisualizerdata	true	Enables or disables saving data for the statistics module in the database.

Table 3 Portal (WS-PGRADE) property for file size to upload

Property name	Default value	Description
wspgrade.maxuploadfilesizeMB	500	Maximum file size in MB to upload into WS-PGRADE portal.

Quota Setting

Another part within service properties is the quota setting. The quota setting consists of two separate property settings: the user quota setting and the repository quota setting. You can set as administrator the user quota in File Storage for limitation of workflow submission per users (Fig. II.9).

Note: if the user just exceeds her/his limit during a submission process, the workflow will be submitted, the submission process won't be interrupted.

The user can save her/his quota by deleting of unuseful workflow instances (in the *Workflow/Concrete/Details* window in WS-PGRADE.) Table 4 shows a File Storage property for enable recalculating user quota to storage.

Table 4 File Storage property for performance improvement

Property name	Default value	Description
guse.storageclient.localmode.sendquota	true	Enables or disable recalculating quota information within the File Storage service. If disabled, file uploads to the Storage service will be much faster, and there will be no quota limits.

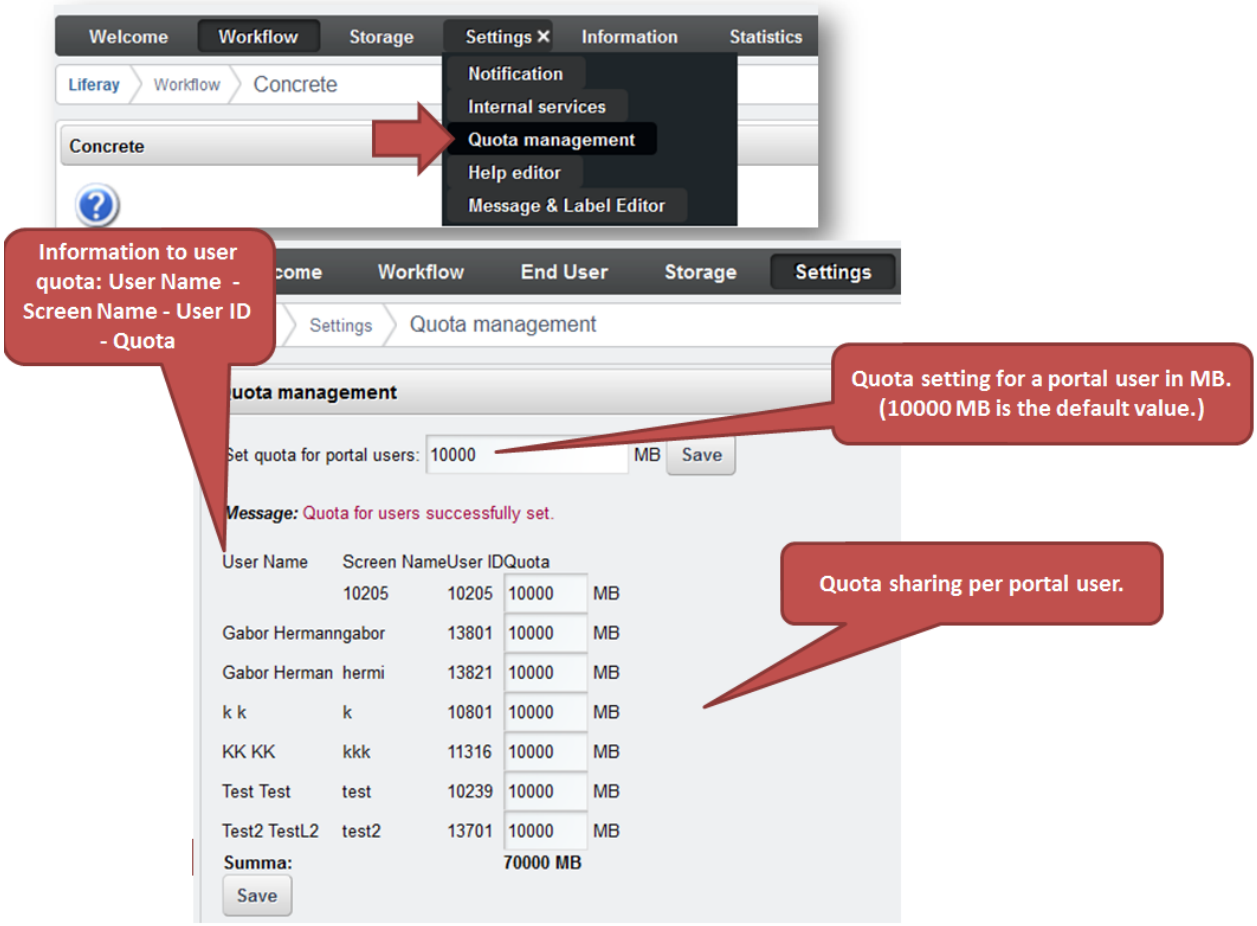


Figure II.9 User quota settings in WS-PGRADE

The second quota type is the repository quota that is related to export function in WS-PGRADE. The repository quota limits the overall size of exported workflows in a local portal repository.

Table 5 Properties for user and repository quota

Setting	Default value	Description
Quota for portal users in MB.	10000	Quota size belonging to a portal user for workflow submission limitation.
Quota for local repository in MB.	5000	Quota size belonging to a portal local repository for workflow export limitation.

The administrator can set the repository quota in the mentioned common place in WS-PGRADE: in the *Settings/Internal services* window. In this case you need to select the *repository* as *Type of Component* then you can set the *quotamax value* through the *Edit* function (see Fig. II.10).

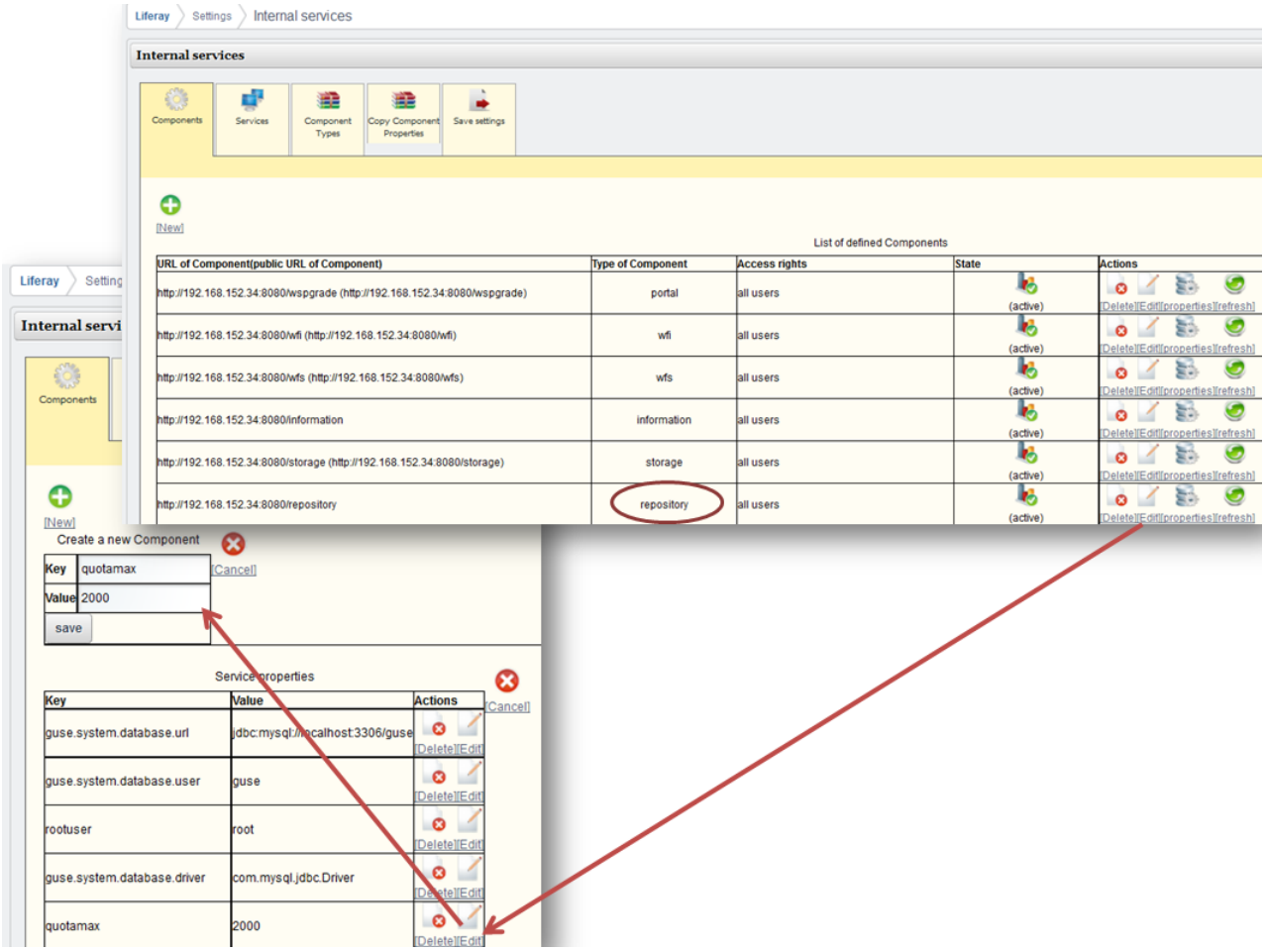


Figure II.10 Repository quota settings in WS-PGRADE

Setting of DCI Bridge Service Properties

You can set the DCI Bridge service properties on the DCI Bridge administrator's interface in the *Middleware settings* function of the selected middleware (Fig. II.11 and Table 6).

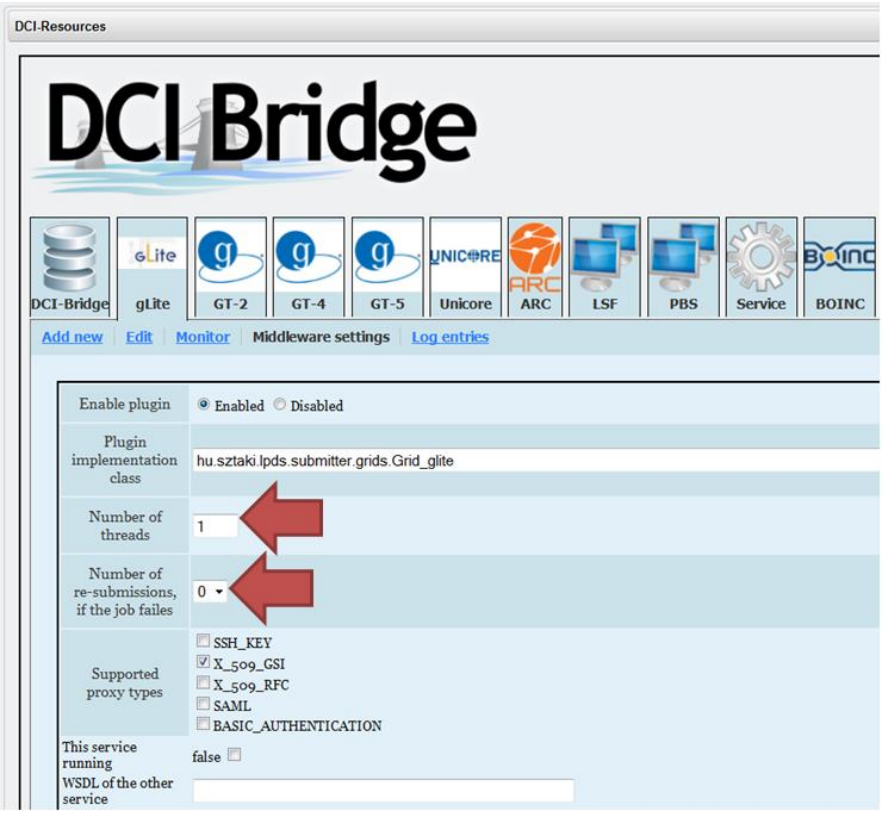


Figure II.11 Properties setting in DCI Bridge administrator's interface

Table 6 DCI Bridge properties

Setting	Default value	Description
Number of threads	1	The given number of threads will manage jobs. The ideal value is between 2-5 in order to speed-up the work of the given DCI plugin. However, too many threads will use up too many resources from the portal server and will slow down the whole portal.
Number of resubmissions	0	The jobs will be resubmitted a given number of times to achieve a successful submission. Since the grid systems are not always reliable resubmission could be useful. However, too many resubmission is useless so the recommended value is 3.

Warning: Once you add a new property or changed an existing property of/to a component, you need to restart in the **Apache Tomcat** administration interface on the **Apache Tomcat Web Application Manager** page the component whose property was changed. (Note: in this page the component referenced as application.

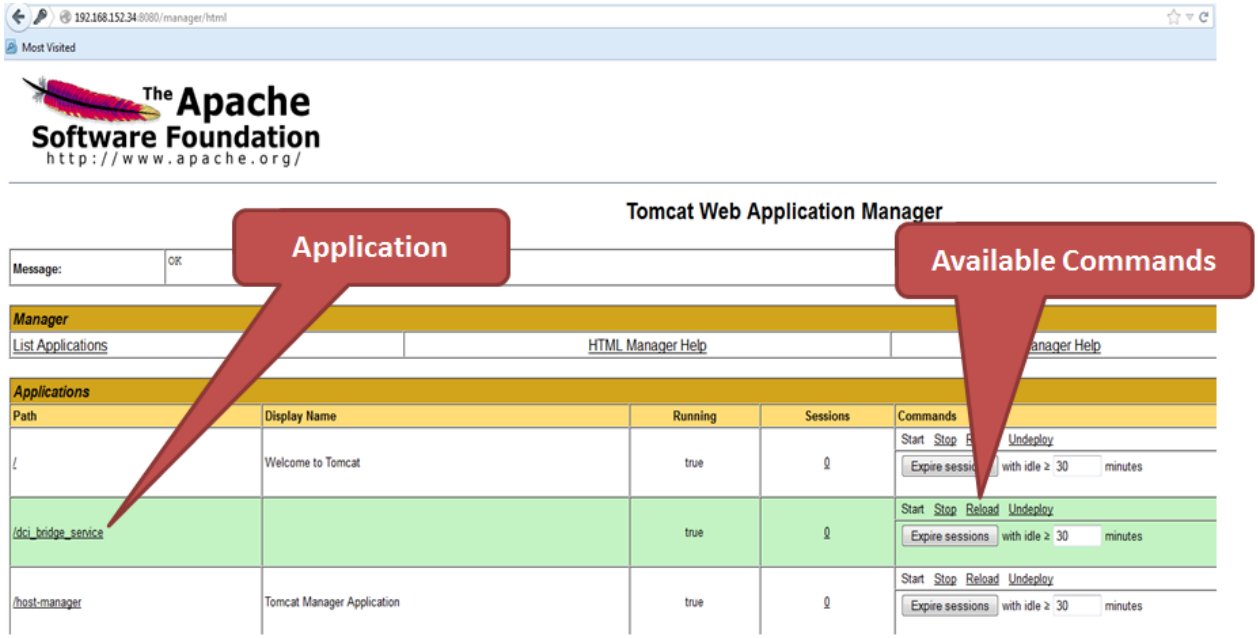


Figure II.12 The Apache Tomcat Web Application Manager page

III. Distinction WS-PGRADE Portal from Liferay Portal

The WS-PGRADE portal is implemented on top of the **Liferay Portal** framework. (Liferay Portal is a widely used free and open source enterprise portal framework.) Liferay is available bundled with a servlet container such as Apache Tomcat that is used as the servlet container to host gUSE services.

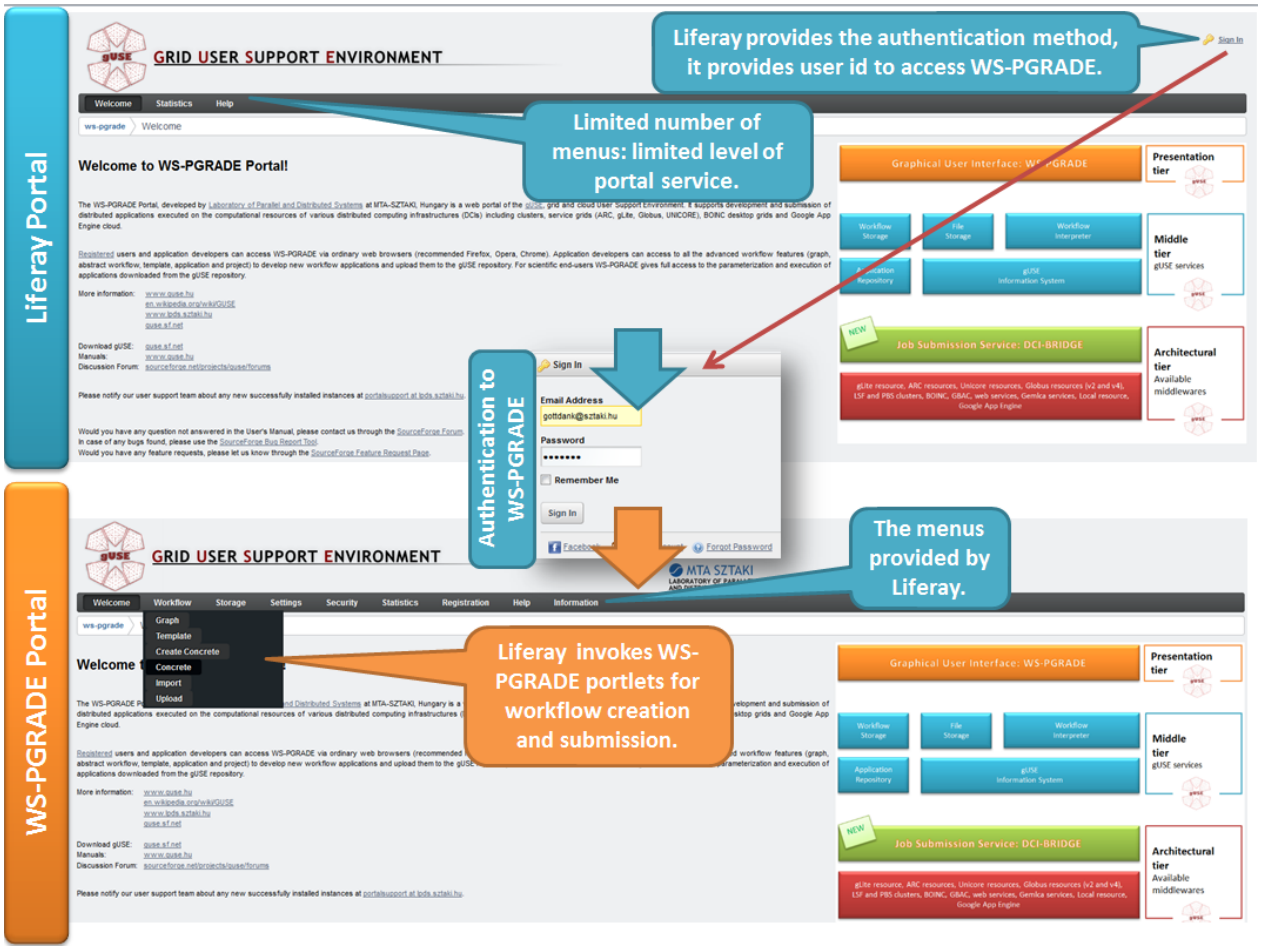


Figure III.1 The Liferay-WS-PGRADE portal comparison

It is important to distinct for many administrative reasons the Liferay Portal environment from the WS-PGRADE portal.

Before you sign in for WS-PGRADE portal you are actually in a Liferay portal. Therefore the sign in authentication method belongs to Liferay and not to WS-PGRADE (upper view in Fig. III.1). Once you signed in as a user you will get a new view where all the WS-PGRADE portlets configured for this portal is shown in the menu. This is already the WS-PGRADE environment

where you can launch WS-PGRADE activities (lower view in Fig. III.1). The essential distinctions are (Fig. III.1):

- Liferay provides authentication service for WS-PGRADE to make file access secure and only accessible to the entitled users.
- Liferay provides the menu structure framework for WS-PGRADE (beside the utility Liferay menu bar, that you can find on the top of WS-PGRADE portal window – see Fig. 14) but the actual menu items come from WS-PGRADE.
- Liferay invokes the WS-PGRADE portlets.
- WS-PGRADE provides the whole gUSE functionality at user interface side.

Thus, Liferay gives the main technology framework of WS-PGRADE. Liferay encapsulates the WS-PGRADE portlets (the whole science gateway functionality) and provides an emerging portal structure together with user authentication solution and menu structure.

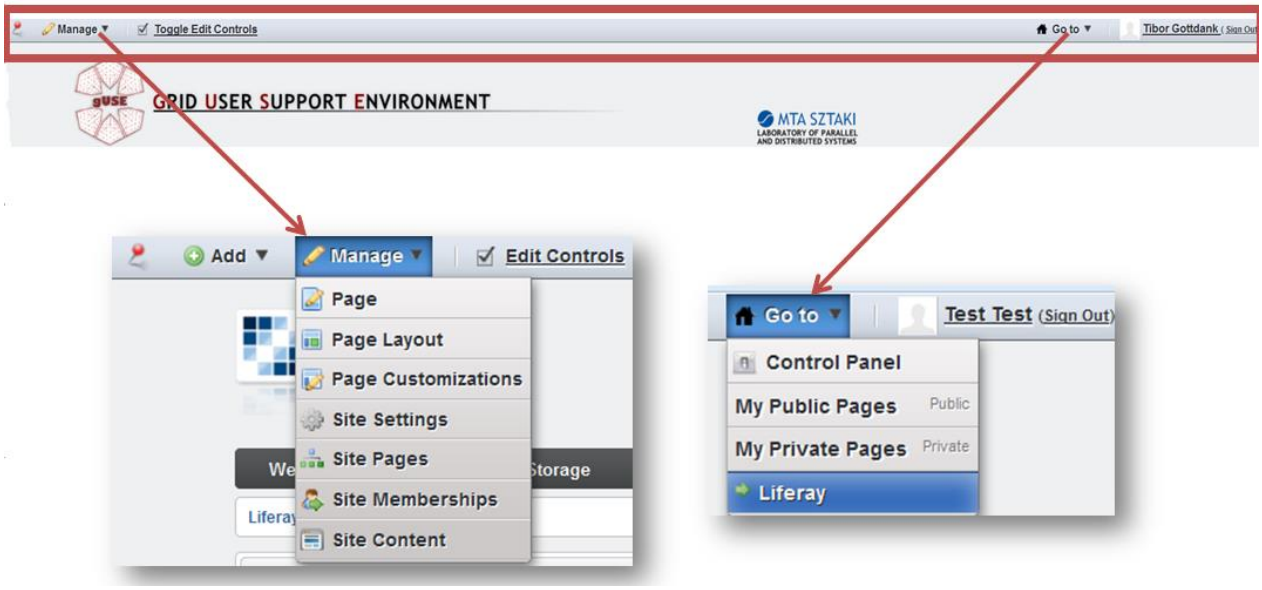


Figure III.2 The Liferay menu bar on the top of the WS-PGRADE/gUSE window and its available content in case of administrator

IV. Setup the End-User Role

Setting of End-User Role

Once you installed the gUSE, you are in **power user**[1] role (or in other words: in **workflow developer user** role). In order to set the **end-user**[2] role as the default role for new users, please do the following Liferay-specific steps:

1. At the top of the portal page, select *Go to/Control Panel* from the Liferay menu bar.
2. At the left panel, click *Portal Settings*.
3. At the right panel, click *Users.*
4. Select the *Default User Associations* menu.
5. Under *Roles*, enter the text "End User" (If you find other user role name remove that)
6. At the right panel, click *Save*. (Fig. IV.1)

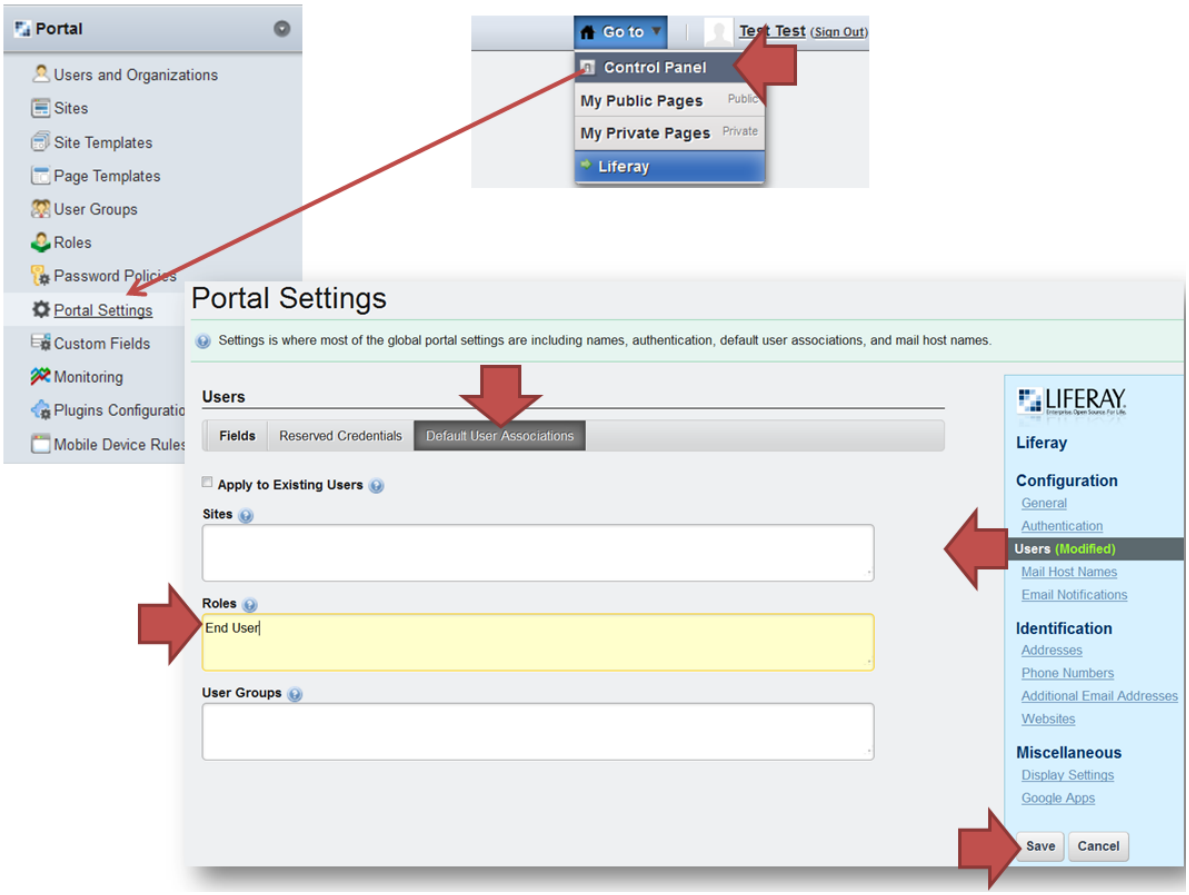


Figure IV.1 End-user role setting in the Liferay-based *Portal Settings* function

[1] The power user builds and configures the workflow applications (either for own use or for the end users).
[2] The end-user has restricted manipulation rights in WS-PGRADE (for example, no access to the workflow definition functions)

This way, new users will receive only the end-user role. You can test your settings by adding a new user: select *Users and Organizations* at the left panel, after click *Add/User*, and fill in the form (see Fig. IV.2).

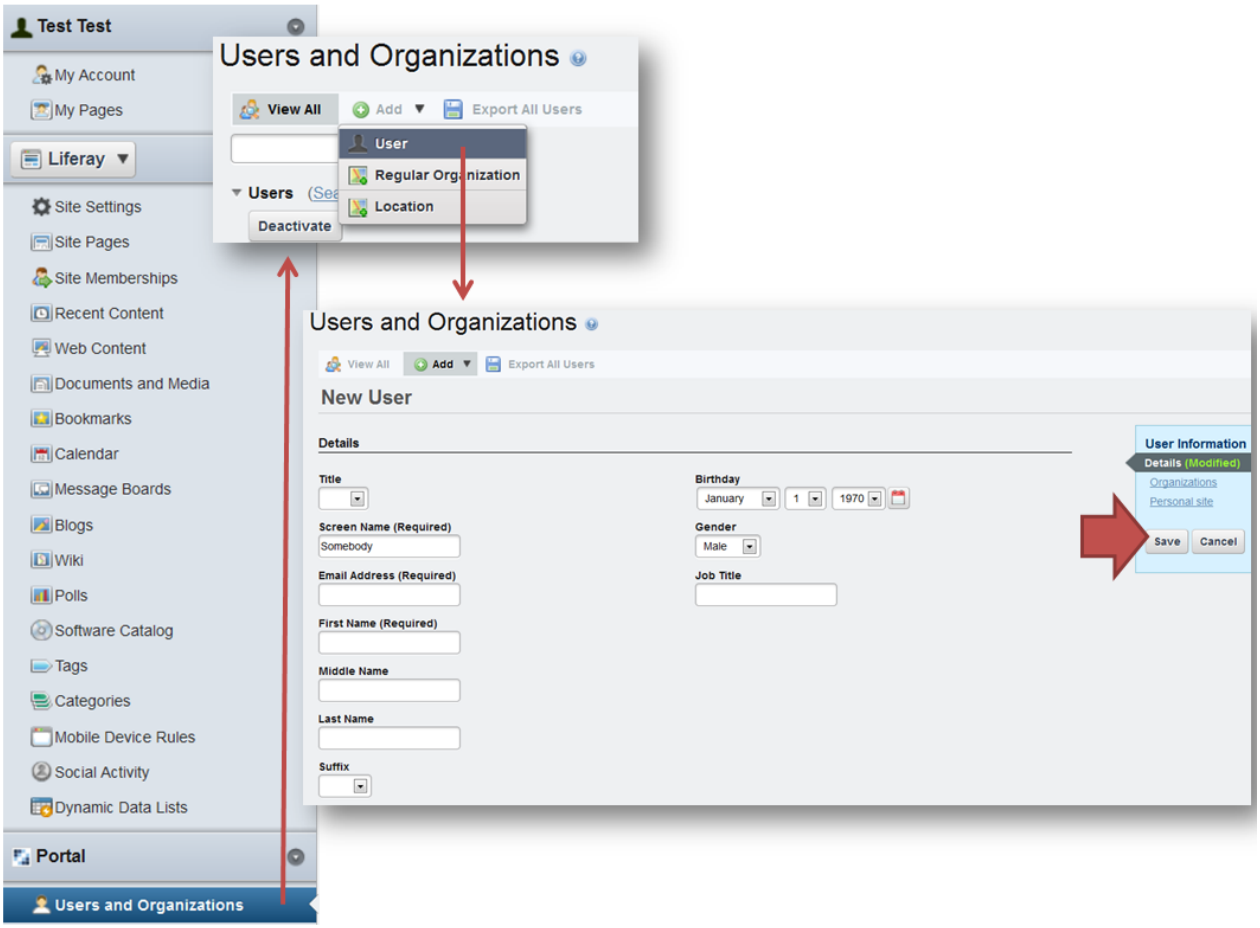


Figure IV.2 Adding a new user in the Liferay-based *User and Organizations* function

Registering the End-User Role

To enable the registration method, do the followings:

1. Within the Liferay *Control Panel* go to the left tab, select *Portal Settings.*
2. At the right tab, select *Authentication.*
3. Under *General* tab, you should see that anybody is allowed to create accounts. (Under the other tabs, you can configure and enable other login methods, e.g. *Facebook* - you need to register a new application under Facebook for this). (Fig. IV.3)

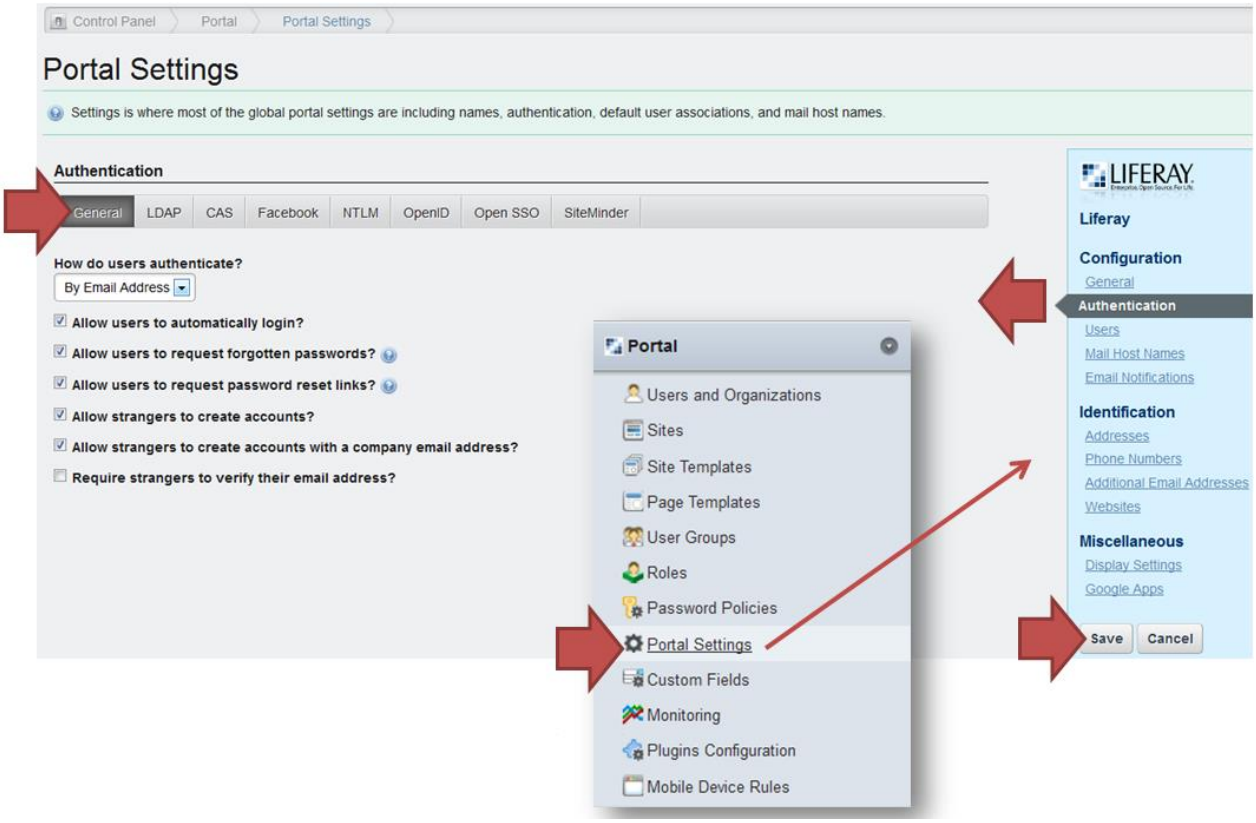


Figure IV.3 Authentication mode settings for end-users in the Liferay-based *Portal Settings/Authentication* function

Menu Visibility Modification

If you don't need in end-user view a menu (in Liferay terminology a site page or in programming terminology a portlet) in the default menu bar, follow the next steps:

1. Let's say, we don't need *Security* from menu bar: Log in as portal administrator.
2. At the top, select *Manage/Site Pages.*
3. In the left part, click *Security.*
4. In the top middle, click on *Permissions.*
5. For the *End User* role, uncheck the *View* permission. Click *Save* (Fig. IV.4)

For checking the setting result, log in as end-user to portal and check the appearing menu structure for end-user: the *Security* menu will invisible (as you set before) and the whole provided functions will reduced for end-user neeeds (Fig. IV.5).

The end-user setup process is in video: http://www.youtube.com/watch?v=rz3KLtO0eds

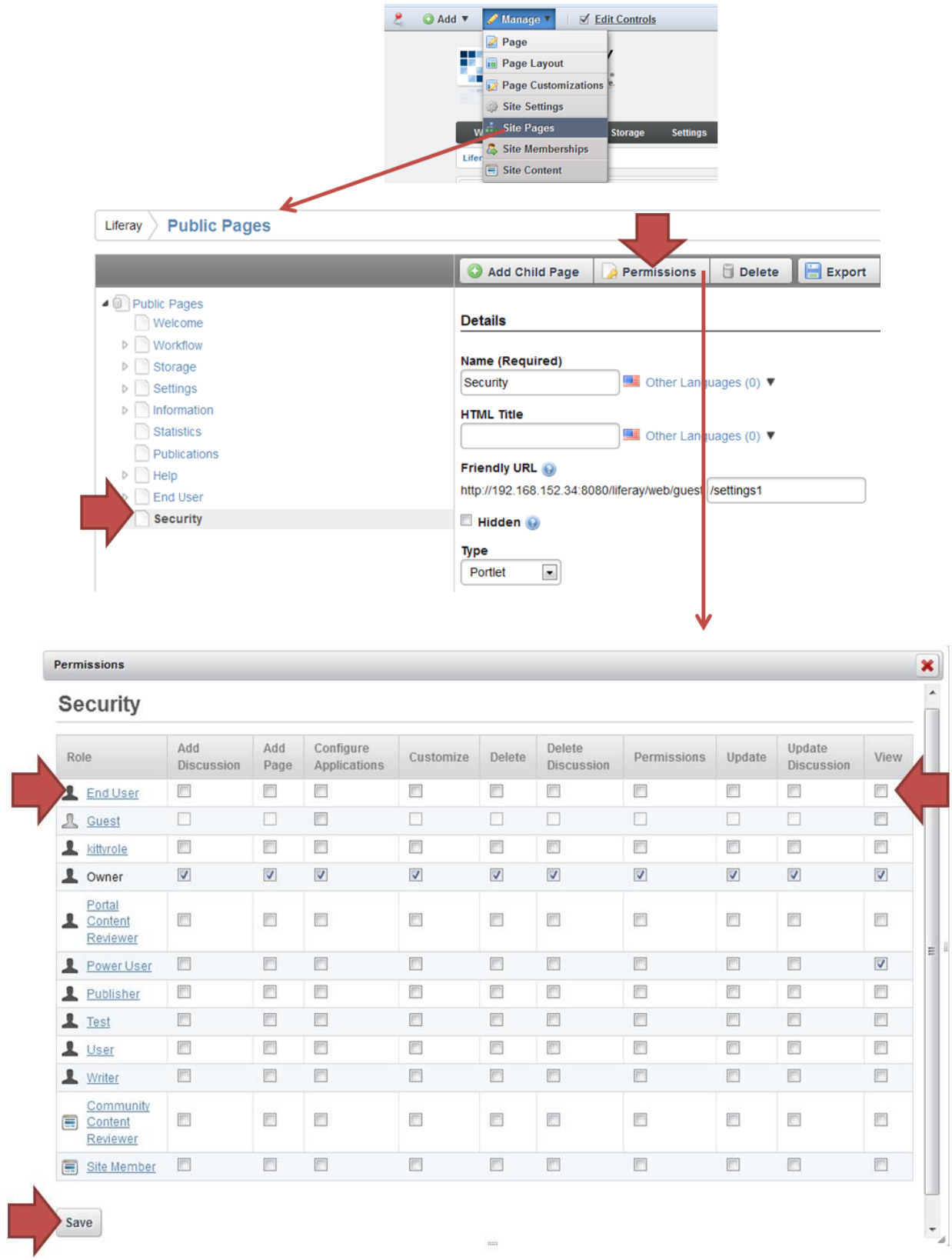


Figure IV.4 Unchecking the *View* permission of the *Security* menu in end-user role in the Liferay-based *Security/Permissions* function

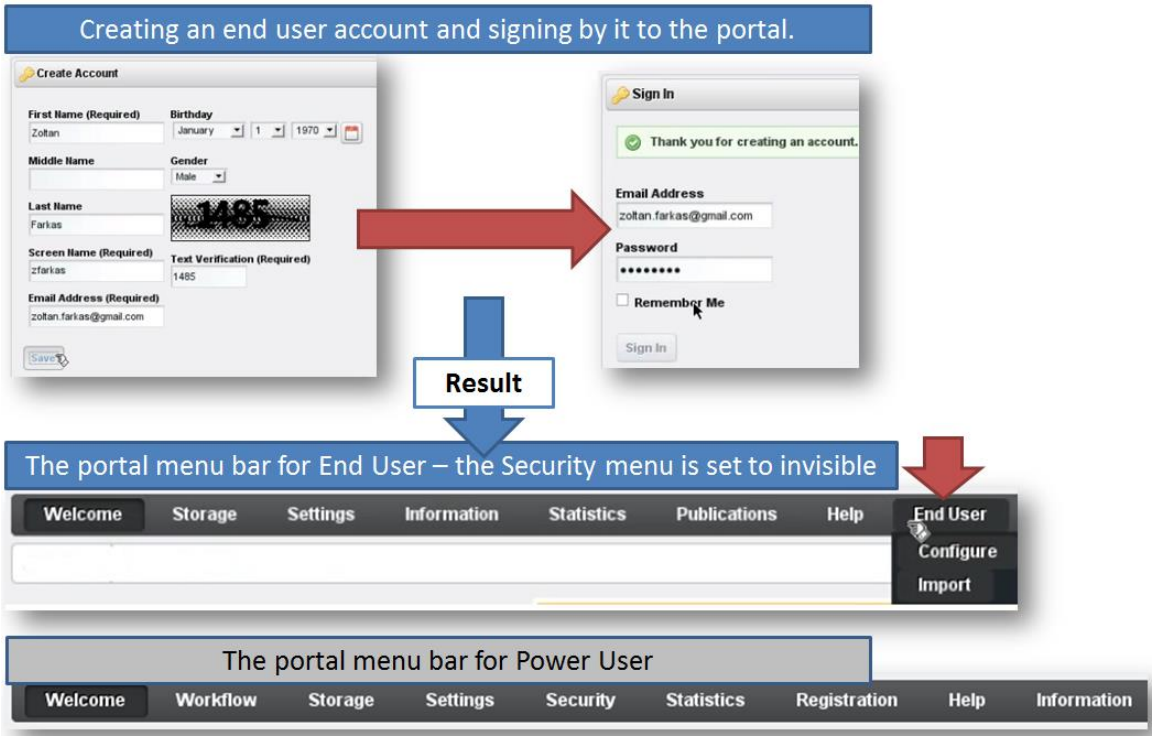


Figure IV.5 Checking the result of the end-user role setup

V. Settings on CloudBroker Platform

In order to use gUSE/WS-PGRADE for job submission to a cloud resource managed by **CloudBroker Platform** (**CBP**), you need to do some preliminary registration and setting operations.

Roles

Basically, there are two types of users in CB environment: administrators and standard users. Administrators typically represent **organizations** and have the potential to create new users (manage or delete) belonging to the same organization, whereas standard users (as well as administrators) can query repository and execute softwares (made public). Administrators can deploy new software that is in "private" status initially. Private status means only administrators can see them at listing the repository and only administrators can run, respectively. Once the software has been tested, i.e. they can really be run on the deployed resource and works as expected, it may be published for use by the organization or by anyone accessing the CB platform by setting software status to "public".

Note: you find in this documentation the administrator-specific description of the CBP handling. The description of CB middleware settings you can see in chapter 2.15 within *DCI Bridge Administrator Manual*. (You find user-specific details in chapter *CloudBroker-based Workflow Configuration and Submission* in the *WS-PGRADE Cookbook* and in chapter 17 in the *Portal User Manual*.)

Registration

To access the CBP you have to own username and password provided by CloudBroker GmbH after registration process – which is used in every communication with the platform.

To registration you need to go to the CB platform registration page: https://platform.cloudbroker.com/registration/new_user_organization. Here you can register a new organization together with your personal account (Fig. V.1).

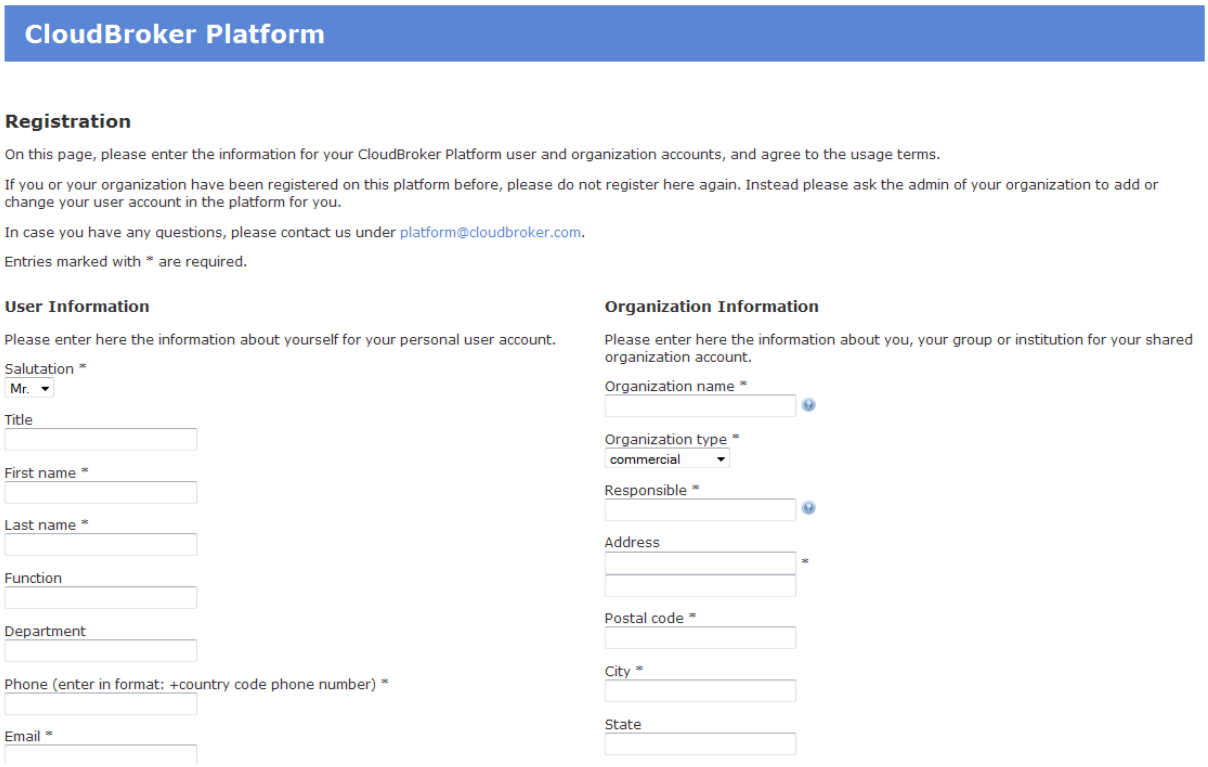

CloudBroker Platform
Registration
On this page, please enter the information for your CloudBroker Platform user and organization accounts, and agree to the usage terms.
If you or your organization have been registered on this platform before, please do not register here again. Instead please ask the admin of your organization to add or change your user account in the platform for you.
In case you have any questions, please contact us under platform@cloudbroker.com.
Entries marked with * are required.
User Information
Please enter here the information about yourself for your personal user account.
Salutation *
Mr.
Title
First name *
Last name *
Function
Department
Phone (enter in format: +country code phone number) *
Email *
Organization Information
Please enter here the information about you, your group or institution for your shared organization account.
Organization name *
Organization type *
commercial
Responsible *
Address
Postal code *
City *
State

Figure V.1 Registration page on CloudBroker Platform

Once you have your user and organization information filled in, read the *Terms of Service* defining the conditions for using the CBP. In case you agree to them, confirm your agreement by clicking on the *I agree* button.

The next step is to familiarize yourself with the platform price conditions. Each organization has a certain monthly payment amount based on the following items:

- Monthly organization cost (“Per month”)
- Monthly cost for each user in the organization (“Per user per month”).

The prices on the above mentioned items may vary depending on the organization type selected in the *Registration* page.

In case the corresponding prices are set to zero, this means that you can register a new user and organization in the platform for free! No platform price conditions page is displayed, and you can directly continue with the registration confirmation.

If a page with CBP price conditions is displayed and you agree to the conditions, click on the *I agree* button to proceed to the final step of your registration.

The final step of the registration process is the registration confirmation. Shortly after that you will receive an email notification that your registration request is being reviewed. Once your registration request has been approved, you will receive a confirmation email. Follow the link in the email to activate your CBP account.

When you click on the activation link, you are directly logged into the platform for the first time. Now that your account is activated, you can log into the platform using your user credentials, i.e. email address and password specified earlier.

After the registration process you will get account to CBP from CloudBroker GmbH and then you can add users to the registered organization.

Let's see the user creation method.

New User Creation

After you login to the CBP (https://platform.cloudbroker.com), you can add users to your organization. For new user adding you do not need to go through the registration process again. You can easily add a new user through the CBP user interface: Simply navigate to the *Users* tab or click on the *Users* link under the *My Organization* section of the dashboard. Once the *Users List* page is opened, click on the *New* button (Fig. V.2).

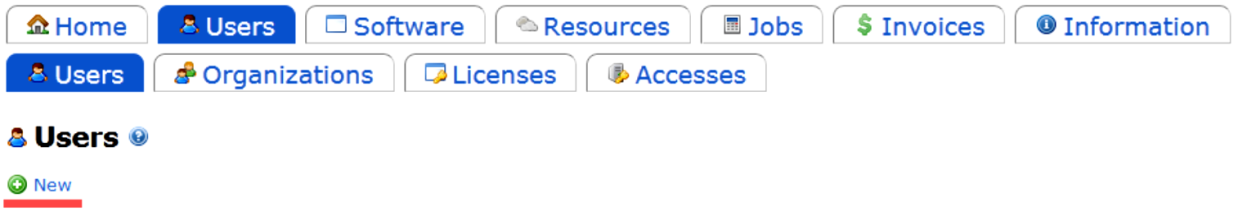


Figure V.2 Users List page

You will see a form for creating a new user with the following mandatory fields on the *Entry Options* sub-sub-tab:

- *Email, Password, Password confirmation, Salutation, First name, Last name, Phone*, and
- *User role* by selecting one of three available user roles: *admin* to create a user with the broadest level of access within the organization,who can
  - create new users
  - run jobs
  - deploy own software
  - register own resources
  - manage accesses/licenses for organization users
  - edit user and organization information

    - handle billing information, invoices and payments
    - have access to everything that the other users in the organization do
- *advanced*, who can
    - run jobs
    - create accesses/licenses for him/herself
    - edit own user information
- *standard*, who can
    - run jobs
    - edit own user information

You can also fill in the optional fields on the *Expert Options* sub-sub-tab. When you have filled in all the necessary user information, click on the *Save* button, okay any cost information if applicable, and a new user will be created.

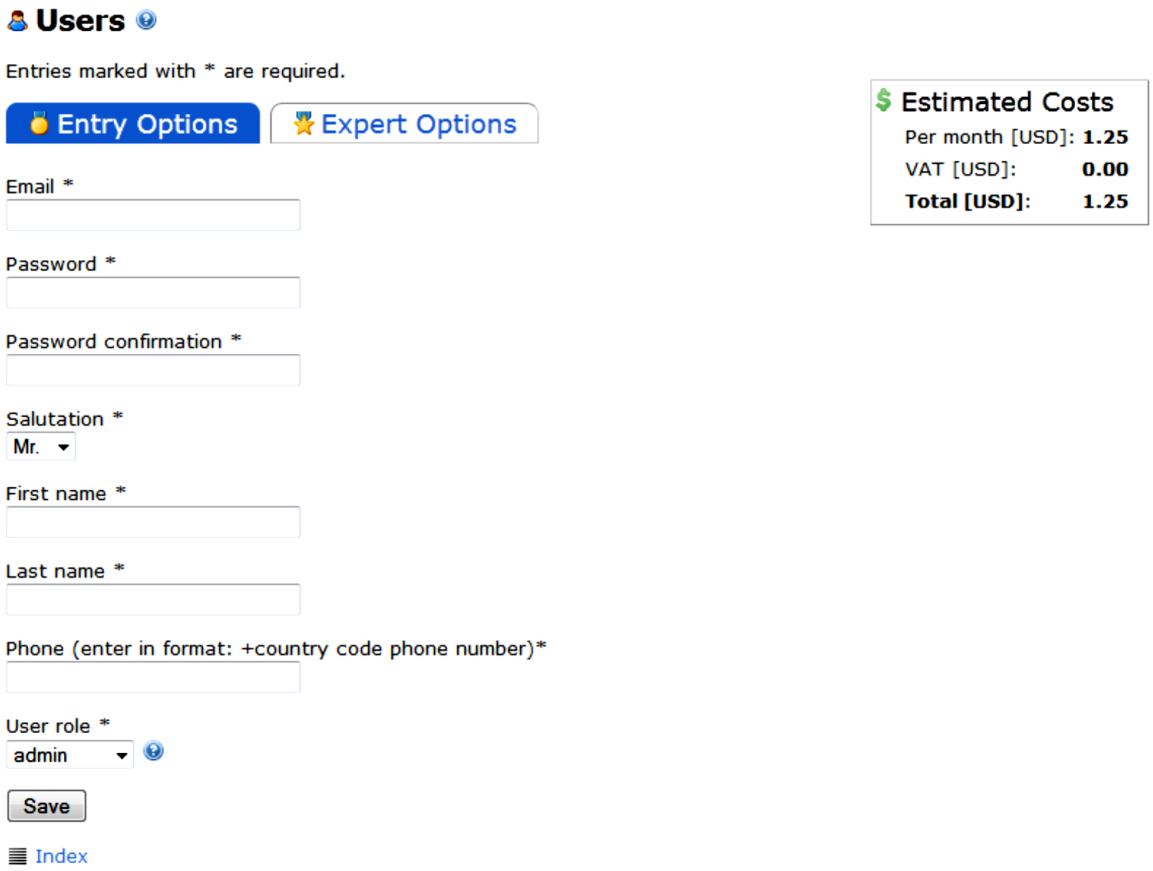


Figure V.3 Create New User page

Now you can register a cloud resource you have access to and add your own application software.

Resource Registration

To use own cloud infrastructures for your applications (e.g. workflow applications configured by gUSE), you need to define the details of this infrastructure (resources, softwares) on a cloud management system.

Then next section introduces the resource registration and software deployment process on CBP.

For *Resource* settings do the followings:

1. Log in to CB Platform site: https://platform.cloudbroker.com (use your valid authentication data that you got after registration process on CB Platform) then click *Resources/New.*
2. In the next window fill the text fields: set *Resource type* (e.g. "*OpenNebula"*), enter *Specifier*, add *Access key ID* and *Secret access key* (Fig. V.4). Note: you get the latter three data not from CBP but from your cloud provider when you

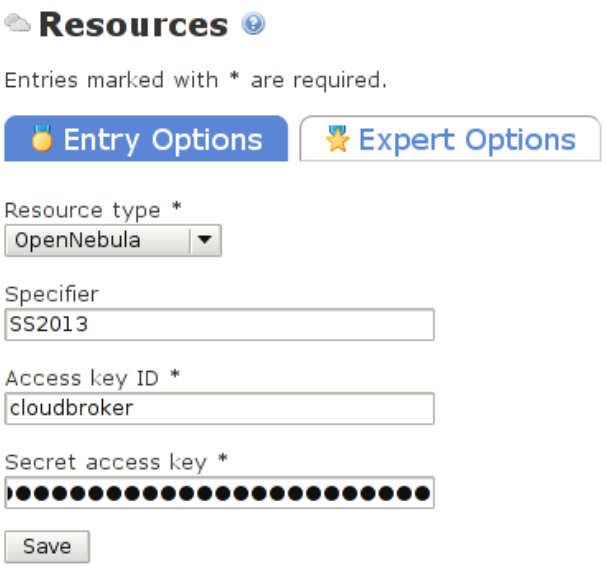


Figure V.4 Resource setting in *Resources/New/Entry Options* page

3. After clicking on *Expert Options* tab set *Maximum nodes*. Set *API URL* (e.g. http://cfe2.lpds.sztaki.hu:4567/)
4. Click on the *Proxy Server Enabled* icon, set proxy access details then click *Save* (see Fig. V.5).

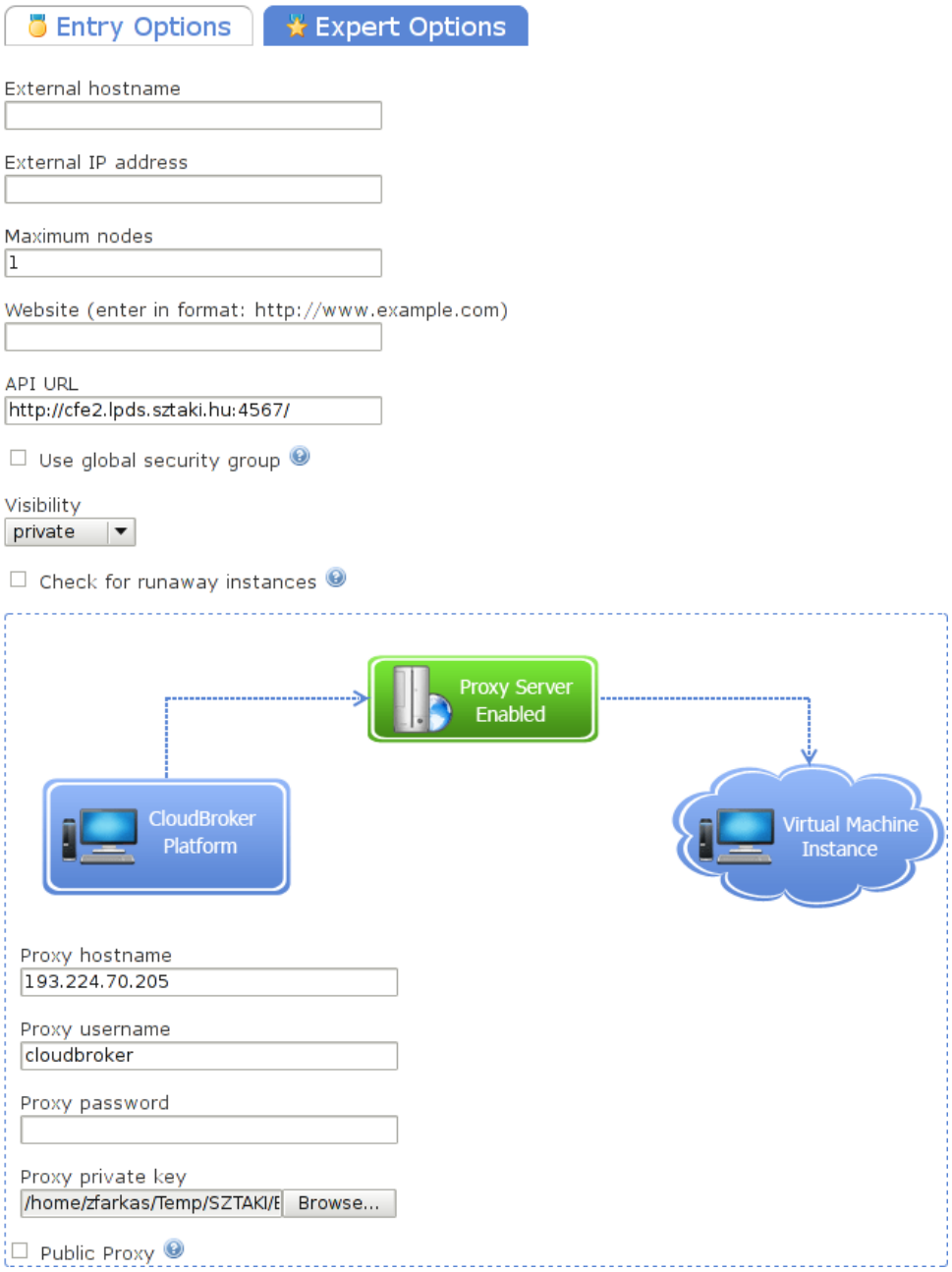


Figure V.5 Resource setting in *Resources/New/Expert Options* page

5. Click the *Instance types* tab then click *New*.
6. In the *Entry Options* tab set *Human readable name* then select the *Resource* as defined earlier. Set *Architecture, CPUs, CPU cores*, *Memory [GB]*, and *API name* (Fig. V.6).
7. Go to *Expert options* tab. Set *Maximum nodes.*
8. Click *Save*.

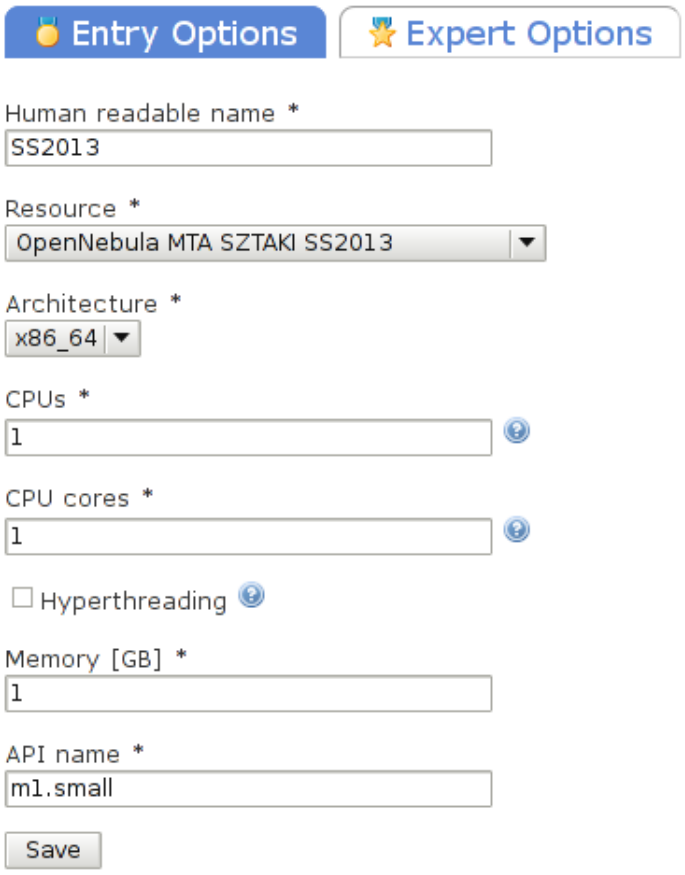


Figure V.6 Instance type setting example in *Instance types/New/Entry Options* page

9. For storage configuration click *Storages/New.*
10. In the *Entry Options* and *Expert Options* tabs configure your storage, then click *Save.*
11. Activate the *Storage* and the *Instance Type* by the *Actions/Activate* function.
12. For region configuration click *Regions/New*.
13. In the *Entry Options* tab set the *Human readable name*, select the *Resource* as defined earlier, set *URL*, select a *Storage* as defined earlier. (Fig. V.7)
14. In the *Expert Options* set *Maximum nodes*
15. Click *Save.*
16. Finally, activate the *Resource* by *Actions/Activate.*

At this point, you have a new resource. Next step: set up new software application.

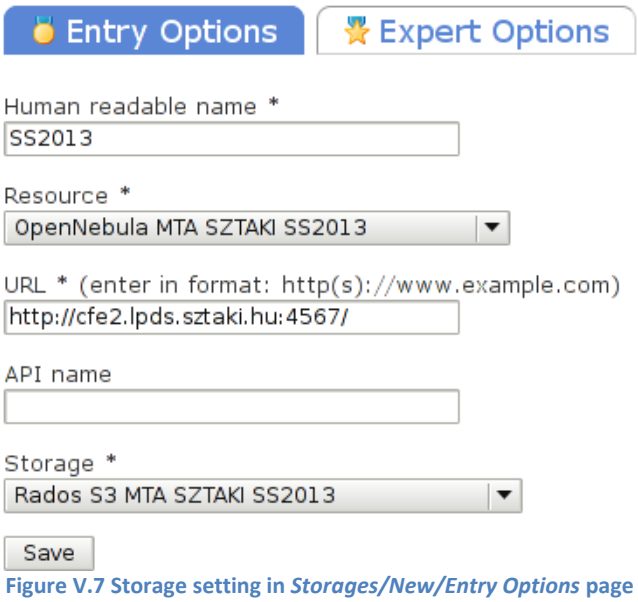


Figure V.7 Storage setting in *Storages/New/Entry Options* page

Software Deployment

After resource configuration deploy your **softwares**. You need to deploy softwares if you don't want to use own executable. (In this case you use the cloud in the SaaS – Software as a Service mode)

Note: The job configuration-specific part of software settings: the user need to select the corresponding software in the *Job Executable* tab job during the job configuration in WS-PGRADE portal.

Let's see the steps:

1. Go to *Software/New*
2. Define the Product and Version details within *Entry Options*. (Fig. V.8)
3. Click *Save*.

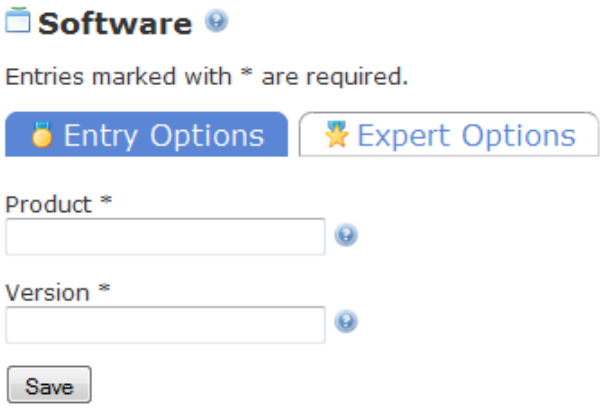


Figure V.8 Software setting in *Storages/New/Entry Options* page

1. To executable definition go to *Software/Executables/New.*
2. Set *Software* and *Binary* within *Entry Options.* (e.g., In case of a PS (Parameter Sweep) workflow called *Autodock* you can deploy the following softwares with binaries)

Software	***Binary***
AutoDock 9 1.0	*ad_generator.sh*
AutoDock 9 1.0	*ad_worker.sh*
AutoDock 9 1.0	*ad_collector.sh*

For every binary you need to perform this operation.

3. Click *Save*.

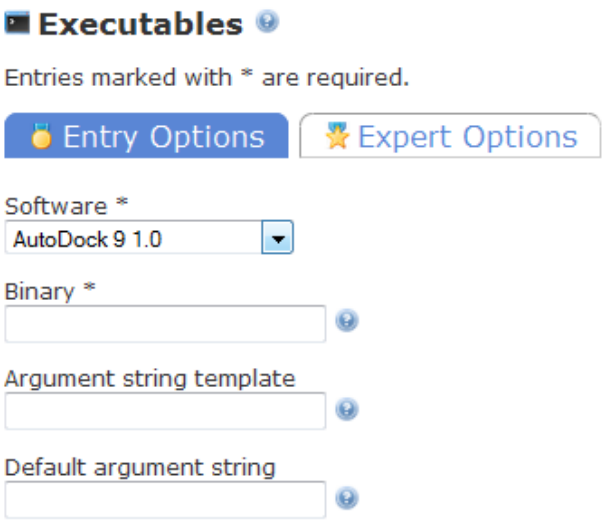


Figure V.9 Executable setting in *Executables/New/Entry Options* page

4. Activate the software under *Software/Actions/Deploy.*
5. Go to *Software/Deployments/New* and set software deployment details: set *Architecture*, *Operating system*, *Installation directory* within *Entry Options*. (see example on Fig. 29)
6. Go to *Expert Options* and set *Image ID* (you got it from your cloud provider) and *username.*
7. Click *Save*. (Perform this action for every software you want.)
8. Activate your *Deployment* and *Software* by the *Actions/Activate* functions on the *Deployments* and *Software* pages.

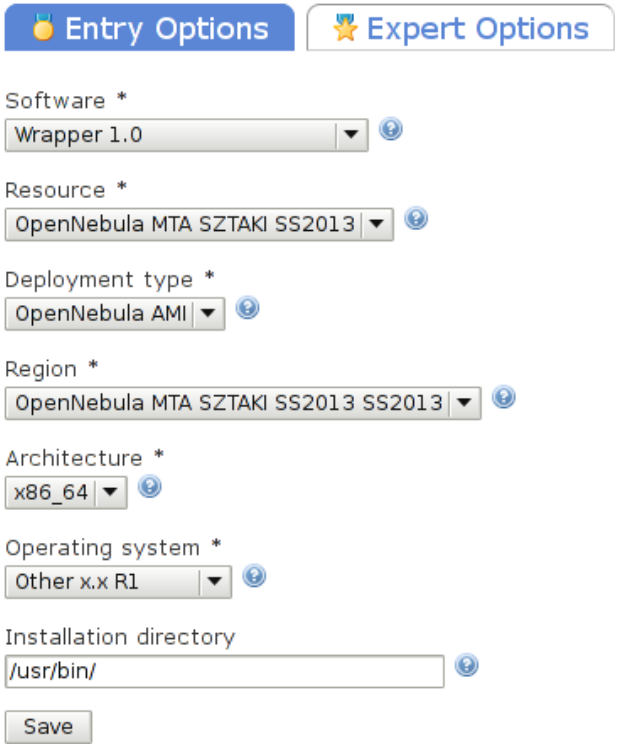


Figure V.10 A software deployment setting example in *Software/Deployment/New/Entry Options* page

Wrapper Deployment

If you want **to use own executables** (in this case you use the cloud in **IaaS - Infrastructure as a Service** mode), then you don't need to register softwares on CBP but you need to deploy a **wrapper** application. The wrapper application is responsible for starting own executable (e.g. *execute.bin*) that is related to the job.

You have to do the operations of section *Software Deployment* to your wrapper application e.g:

Wrapper 9 1.0	*guse_wrapper.sh*

Note: The job configuration-specific part of wrapper settings: the user need to click on the *Enable own executable* button in the *Job Executable* tab job during the job configuration in WS-PGRADE portal to add wrapper to the corresponding job.

- Resource registration: http://www.lpds.sztaki.hu/summerschool2013/downloads/01%20-%20CloudBroker%20Resource%20Configuration.ppt
- Software deployment: http://www.lpds.sztaki.hu/summerschool2013/downloads/02%20-%20CloudBroker%20Software%20Configuration%20and%20Execution.ppt

Additional information:

- Resource registration: https://platform.cloudbroker.com/documents/CloudBrokerPlatform_ResourceRegistrationGuide-1.2.pdf
- Software deployment: https://platform.cloudbroker.com/documents/CloudBrokerPlatform_SoftwareDeploymentGuide-1.2.pdf

- Resource registration: https://www.lpds.sztaki.hu/services/sw/download.php?download=5384800b18ddc7ecafe9943e2bc13d55
- Software deployment: https://www.lpds.sztaki.hu/services/sw/download.php?download=101c0799689ade585a8c4ca87253cf0d
- Wrapper execution: https://www.lpds.sztaki.hu/services/sw/download.php?download=1bf395604315dc4ae8a9a04b26cf89cc

VI. Ticket Request to Use Data Avenue

The **Data Avenue** is a file commander tool for data transfer, enabling easy data moving between various storages services (such as grid, cloud, cluster, supercomputers) by various protocols: HTTP, HTTPS, SFTP, GSIFTP, S3 and SRM. The Data Avenue is integrated into WS-PGRADE portal as a portlet.

To use Data Avenue services from WS-PGRADE portal, you (as administrator) need a so called **ticket,** which is a kind of access code similar to API key used to access e.g. Google or Amazon services.

The next section describes the ticket requesting and receiving step-by-step process:

1. Once you sign as admin into gUSE/WS-PGRADE portal, select the *Data Avenue* menu.
2. Select the (*Options*) icon on the top right in the Data Avenue window and select *Preferences* in the appearing submenu list.
3. In the next window you find two text fields (Fig. VII.1):
   - The first field is to define the Data Avenue service description access in form of URL. The default value is the official access of Data Avenue WSDL (applying the default value is recommended).
   - In the second field you need to add the *portal administrator ticket*. You can get it by the next procedure:
     - Click on the *Ticket request form* link at the bottom right side (see Fig. VII.1). The *Data Avenue Ticket Request Form* will appear (Fig. VII.2).
     - Fill the request form (mandatory to fill: First name, Last name, and E-mail address as well as the CAPTCHA test field). If you want to be on *Data Avenue Users Map* (drag the marker if the estimated location is imprecise). If you use Data Avenue in WS-PGRADE portal, then select the *Portlet* radio button instead of *API* option. (However, if you want to access Data Avenue from API instead of using gUSE/WS-PGRADE-based portal, then choose the option *API.*)
     - Click on *Submit*.

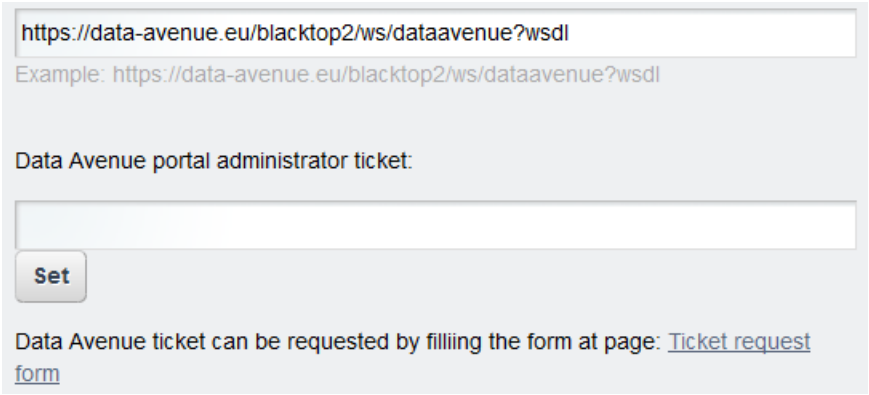

https://data-avenue.eu/blacktop2/ws/dataavenue?wsdl
Example: https://data-avenue.eu/blacktop2/ws/dataavenue?wsdl
Data Avenue portal administrator ticket:
Set
Data Avenue ticket can be requested by filliing the form at page: Ticket request form

Figure VI.1 Setting the portal administrator ticket

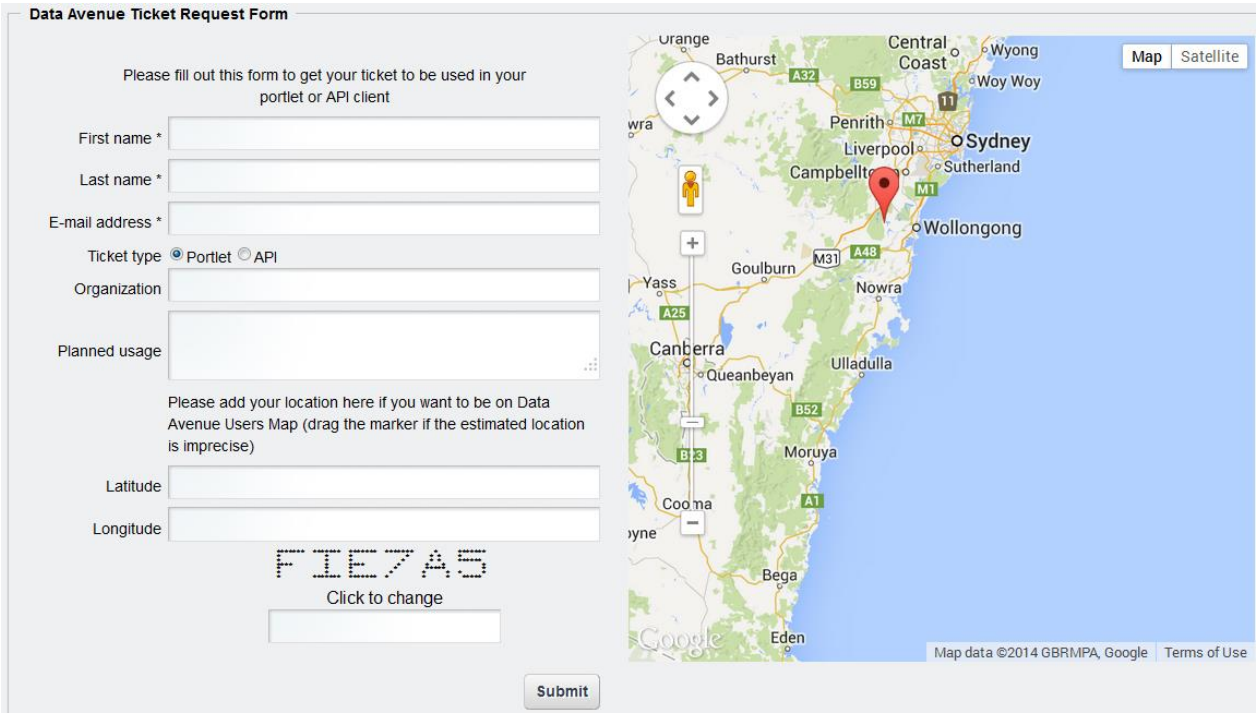


Figure VI.2 The ticket request form

4. A short time after you will get an automatically generated email about receiving your ticket request. Your request will be processed in maximum two-three days and you will get your ticket by email. Once you received your ticket, type it into the *Data Avenue portal administrator* ticket field (Fig. VII.1).

Note: The validity period of a ticket is **one year**. The validity can be extended after expiration of ticket (you need to send a new ticket request).

5. Click on *Set*. The message *Data has been saved* will appear. Click on *Ok*.
6. From this point Data Avenue is available for every user who uses this portal.

VII. Embedding SHIWA Repository into WS-PGRADE

The goal of the Liferay-based configuration, which is mentioned in this chapter, is to browse and select submittable workflows (in WS-PGRADE terminology **Submittable Execution Nodes - SENs**) located in the SHIWA Workflow Repository from WS-PGRADE.

For embedding, please follow the next steps:

1. Create new page for **SHIWA Repository:** in Liferay toolbar of your WS-PGRADE portal, click *Add* in the top menu, then click *Page* submenu (Fig. VIII.1).

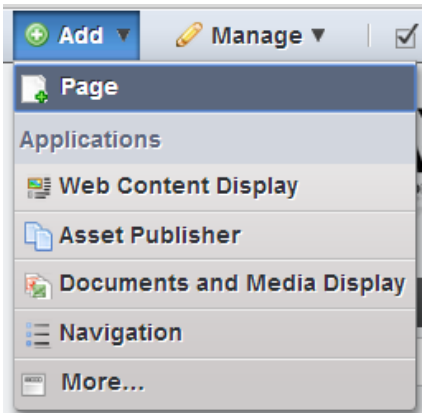


Figure VII.1 The Add/Page menu in Liferay toolbar

2. Type a name for the menu (e.g. *SHIWA Repository*). Click on the *SHIWA Repository* menu that is created.
3. Click on the same *Add* button in the top menu then click *More*. (List of the application groups will pop up in the left hand-side.)
4. Open *Sample* group then click to *add* in the row of *IFrame* (Fig. VIII.2) A new portlet will be generated in the page.

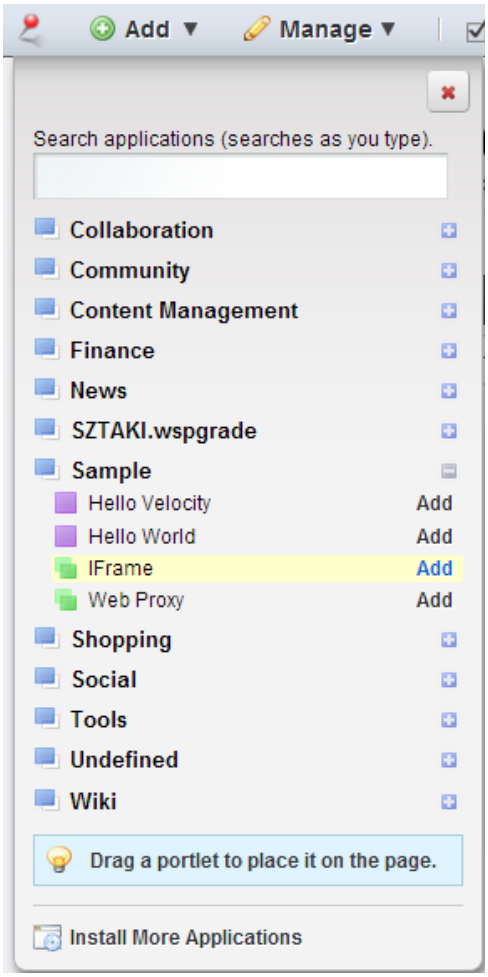


Figure VII.2 Adding IFrame in Liferay toolbar

5. Click on the *"Please configure this portlet to make it visible to all users."* link and fill in the boxes as it is shown in Fig. VIII.3. Properties necessary to fill:
   - *SourceURL*, the URL of SHIWA Repository
   - *Field Name*: *sspUserId* with the value: *@user_id@*
   - *Hidden variable*: *sspServiceId*=XXXXX, where XXXXX is the password that you have got from the administrator of the SHIWA Repository.

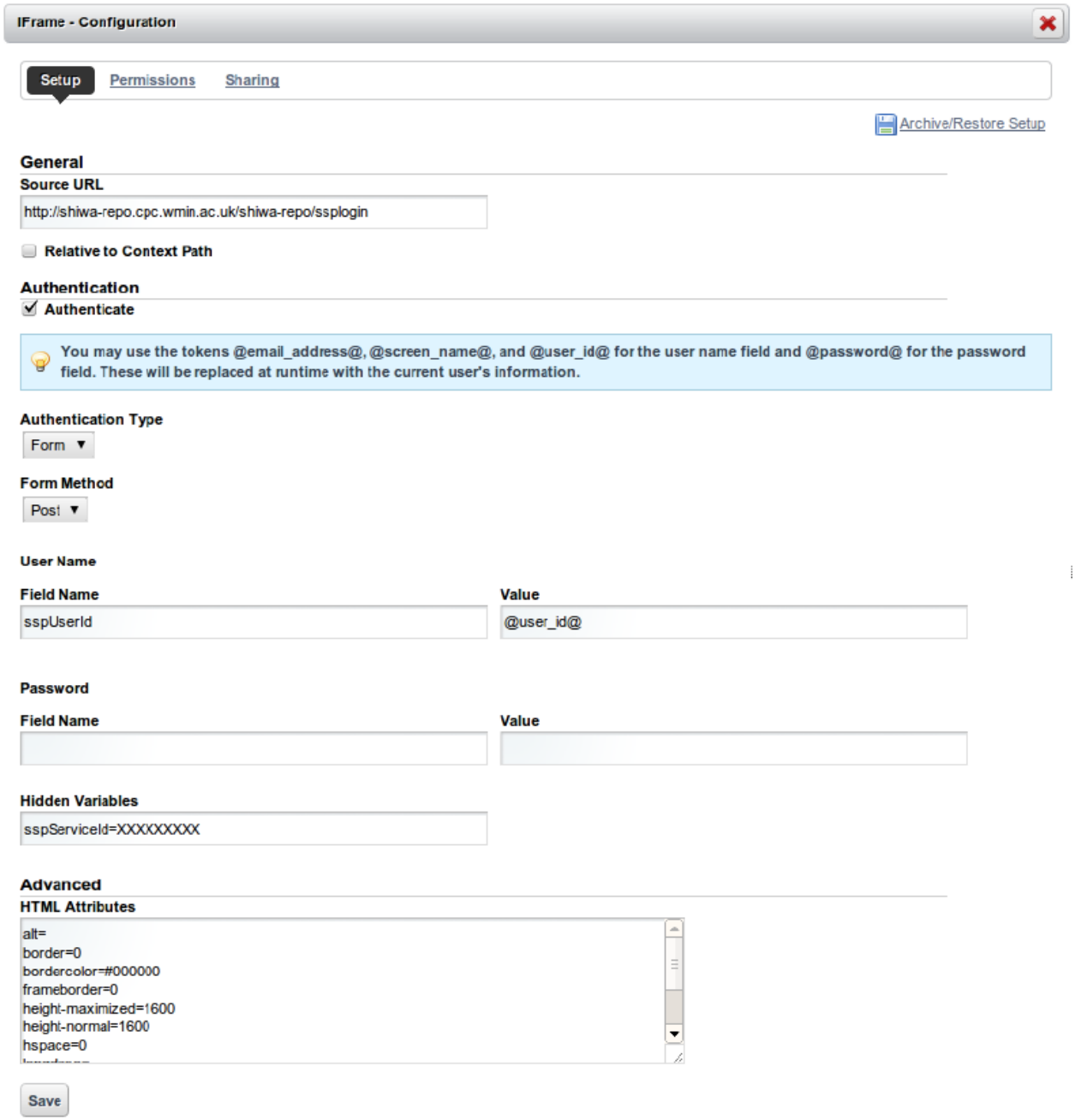


Figure VII.3 Configuring IFrame

6. Click *Save* button. From this point you can access by the *SHIWA Repository* menu the SHIWA Workflow Repository site and the workflows are searchable in the *Find Workflows* field as shown in Fig. VIII.4.

Figure VII.4 Embedded SHIWA Repository

VIII. Additional Settings in the Case of Untrusted Certification

If you add **HTTPS-type** remote URLs (e.g. within middleware settings in DCI Bridge or within input source definition in WS-PGRADE) instead of HTTP-type URLs, make sure of valid certification path (e.g. by a browser).

If the certificate is not trusted (e.g. your browser shows you an error message about that), you need to add a **trusted keystore** by Java.

The next description[3] explains the necessary steps as well as you find example how you can do it in case of **SHIWA** resource (this specific parts will be in purple in the text). **The relevant parts of the console outputs are in dark blue**.

Generic problem: there are a configured Tomcat supported SSL and a deployed web service on a development Tomcat server. During the connection to the deployed web service over SSL via this URL: **https://localhost:8443/HelloWorld/hello?wsdl**, it hits:

```
javax.net.ssl.SSLHandshakeException:

   sun.security.validator.ValidatorException: PKIX path building failed:

   sun.security.provider.certpath.SunCertPathBuilderException:

   unable to find valid certification path to requested target


Caused by: sun.security.validator.ValidatorException:

   PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException:

   unable to find valid certification path to requested target


Caused by: sun.security.provider.certpath.SunCertPathBuilderException:

   unable to find valid certification path to requested target
```

The steps of solution:

1. Get **InstallCert:** Get **InstallCert** class from http://opentox.ntua.gr/files/InstallCert.zip (download and then unzip).

[3] The text based on the "Unable To Find Valid Certification Path To Requested Target" article: http://www.mkyong.com/webservices/jax-ws/suncertpathbuilderexception-unable-to-find-valid-certification-path-to-requested-target/

2. Add **Trusted Keystore:** Run **InstallCert**, with your hostname and HTTP port, and press "**1**" when ask for input. It will add your "**localhost**" as a trusted keystore, and generate a file named "**jssecacerts**":

```
java InstallCert localhost:8443
```

In case of SHIWA-based middleware settings in DCI Bridge you have to add two executables: for **SHIWA Submission Service URL** and for **SHIWA Workflow Repository** (instead of `java InstallCert localhost:8443`)

An example:

```
java InstallCert submission.cpc.wmin.ac.uk

java InstallCert shiwa-repo.cpc.wmin.ac.uk
```

You will get an output like this:

```
Loading KeyStore C:\Program Files\Java\jre6\lib\security\cacerts...

Opening connection to localhost:8443...

Starting SSL handshake...

javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException:
PKIX path building failed: sun.security.

provider.certpath.SunCertPathBuilderException: unable to find valid
certification path to requested target

  ...

Server sent 1 certificate(s):

1 Subject CN=yong mook kim, OU=mkyong, O=mkyong, L=puchong, ST=PJ, C=my

Issuer  CN=yong mook kim, OU=mkyong, O=mkyong, L=puchong, ST=PJ, C=my

sha1    32 3e 15 42 96 ba e9 4d 9c 5d e7 5e 6b 0f 30 23 b4 e3 f4 98

md5     c8 dd a1 af 9f 55 a0 7f 6e 98 10 de 8c 63 1b a5
```

At this point you need to add "**1**":

```
Enter certificate to add to trusted keystore or 'q' to quit: [1]

1

[

[

 Version: V3
```

```
Subject: CN=yong mook kim, OU=mkyong, O=mkyong, L=puchong, ST=PJ, C=my

Signature Algorithm: SHA1withRSA, OID = 1.2.840.113549.1.1.5

Key:  Sun RSA public key, 1024 bits

     ...

     ]

     ...

     ]
```

Added certificate to keystore 'jssecacerts' using alias 'localhost-1'

3. **Verify Trusted Keystore:** Try run the **InstallCert** command(s) again (in case of SHIWA: **java InstallCert submission.cpc.wmin.ac.uk** and **java InstallCert repo-test.cpc.wmin.ac.uk)**
   The connection should be OK now.

   ```
   java InstallCert localhost:8443
   ```

Thus, in case of SHIWA use the already mentioned two commands (instead of `java InstallCert localhost:8443`).

An example:

```
java InstallCert submission.cpc.wmin.ac.uk

java InstallCert repo-test.cpc.wmin.ac.uk
```

Output:

```
Loading KeyStore jssecacerts...

Opening connection to localhost:8443...

Starting SSL handshake...
```

No errors, certificate is already trusted

```
Server sent 1 certificate(s):

1 Subject CN=yong mook kim, OU=mkyong, O=mkyong, L=puchong, ST=PJ, C=my

Issuer CN=yong mook kim, OU=mkyong, O=mkyong, L=puchong, ST=PJ, C=my

sha1    32 3e 15 42 96 ba e9 4d 9c 5d e7 5e 6b 0f 30 23 b4 e3 f4 98

md5     c8 dd a1 af 9f 55 a0 7f 6e 98 10 de 8c 63 1b a5
```

At this point you can quit by "**q**":

```
Enter certificate to add to trusted keystore or 'q' to quit:
[1]

q

KeyStore not changed
```

4. Copy **jssecacerts:** Copy the generated **jssecacerts** file to your **$JAVA_HOME\jre\lib\security** folder.
5. Restart Tomcat.

IX. Administration of Single Job Wizard

Single Job Wizard simplifies workflow creation and execution in the simplest case, if the workflow contains one job only. What looks a huge disadvantage in aspect of the possibilities of workflow development provides more advantages to create and execute these kinds of workflows: no Workflow Editor required, several creation and configuration interfaces are not needed in this case anymore. The job will be executed on those resource what is prepared by the portal administrator, so if mechanism of the robot certificates are utilized, then the users do not have to take care about authentication neither. The users are able to drop their applications, and the inputs required then specify the names of the output files going to be generated by the application. And that's all, folks, the job is going to be submitted to the resource.

In technical terms Single Job Wizard is based on **ASM API** provided by WS-PGRADE/gUSE system. In its heart a prepared one-job workflow is staying containing 4 inputs(the application, inputs compressed, command line arguments, output names descriptor file) and 1 output(all the required outputs compressed). It executes a simple script that is responsible for the decompression of the inputs and executes the application. Once the application is finished successfully, it gets the certain output files specified by the user, and compress them and it ends. If the application fails the wrapper script inherits in exit code and returns. Once the core system realizes that the wrapper is finished, it downloads the output (the compressed file) and set the proper status for the application.

Single Job Wizard Setup

For a successful setup and configuration you must follow the next steps:

1. **Deploy the web application**: Deploy the web application (**singlejobwizard.war**) through Liferay by clicking to **Go to** -> **Control Panel** -> "**Plugins Installation**" in the "**Server**" group in menu on the left-hand side.
2. **Registrate it as a new component in Internal Services**: Navigate to **Internal Services** interface in **Settings** menu, and add a new component with the following properties:
   - **Type of Component**: portal
   - **Service group**: gUSE
   - **URL of Component**: PORTAL_URL/singlejobwizard
   - **URL to initialize Component**: PORTAL_URL/singlejobwizard/init
   - **Public URL of Component**: PORTAL_URL/singlejobwizard
   - **State**: active

   In the same interface, go to "**Copy Component properties**" panel and copy the properties of the **wspgrade** component to **singlejobwizard** component.

3. Restart the portal: get the process ID of java by executing the following command:
   ```
   JAVAPID=`ps aux | grep java`
   ```
   then kill the process:
   ```
   kill -9 $JAVAPID
   ```
   Finally start the portal by executing startup.sh located in tomcat's bin folder:
   ```
   ./startup.sh
   ```

4. **Initialize it and add new page**: Once the portal has been started successfully, add new page (Optional) by clicking "**Add**" button on the top menu and clicking to "**Page**" button (see next figure) and then by typing its name. The settings can be finalized by clicking the green tick. Associate the portlet to the page by clicking to "Add" button on the top menu and then selecting option "More". Then, after the categories of portlets appear, open SZTAKI.wspgrade category and click "**add**" button in the row of Single-Job Wizard.
5. **Upload, reconfigure and export the workflow-skeleton provided**: You can download the skeleton workflow from here. Go to upload portlet and upload the prepared workflow to the portal server. Then reconfigure its resource adjusted (by default is local submitter, for production purposes CloudBroker + Robot Certificate is suggested). Finally, export the reconfigured workflow as application into the **Local Repository** by clicking **Export** button.

Configuration

1. Open the portlet's configuration panel by clicking to the wrench of the top right corner of the portlet (Fig. X.1).

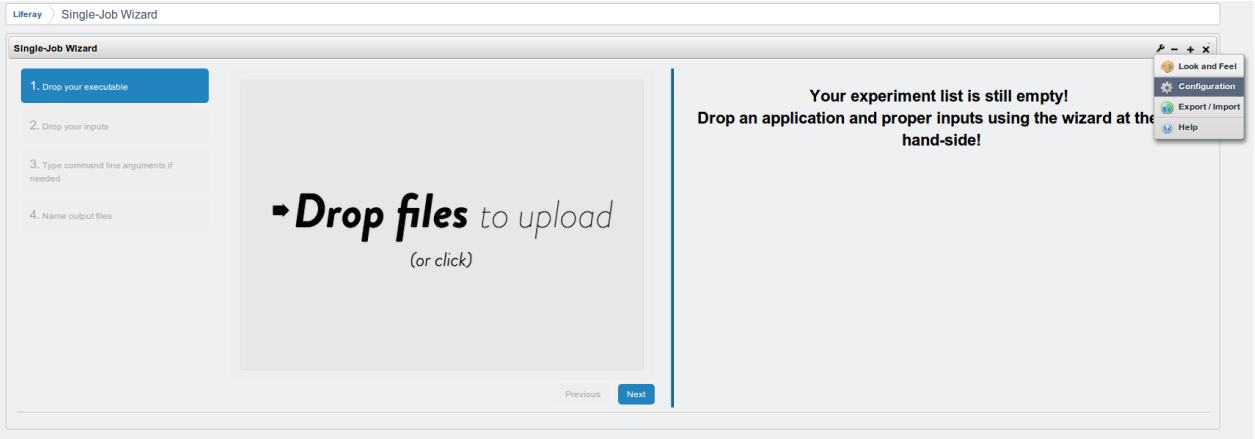


Figure IX.1 Single Job Wizard configuration 1.

2. Then select the workflow that you have exported as a base workflow for Single Job Wizard.

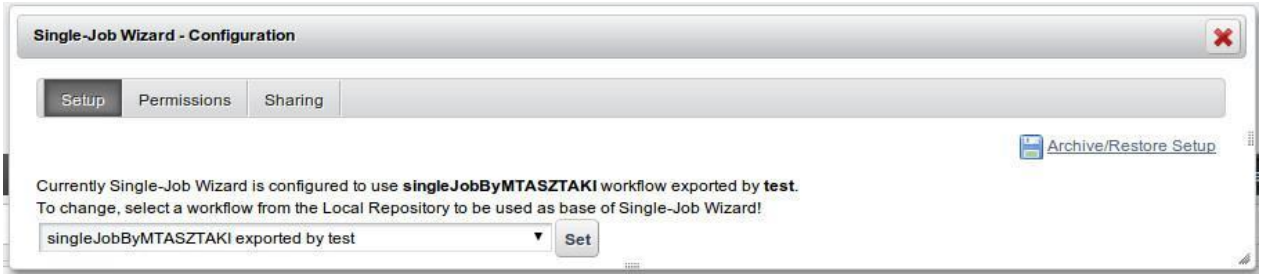

Single-Job Wizard - Configuration
Setup
Permissions
Sharing
Archive/Restore Setup
Currently Single-Job Wizard is configured to use singleJobByMTASZTAKI workflow exported by test.
To change, select a workflow from the Local Repository to be used as base of Single-Job Wizard!
singleJobByMTASZTAKI exported by test
Set

Figure IX.2 Single Job Wizard configuration 2.

3. Finally, set permissions in "**Permissions**" panel to hide the wrench from the users avoiding reconfiguration and to allow **View** option for them (Fig. X.3).

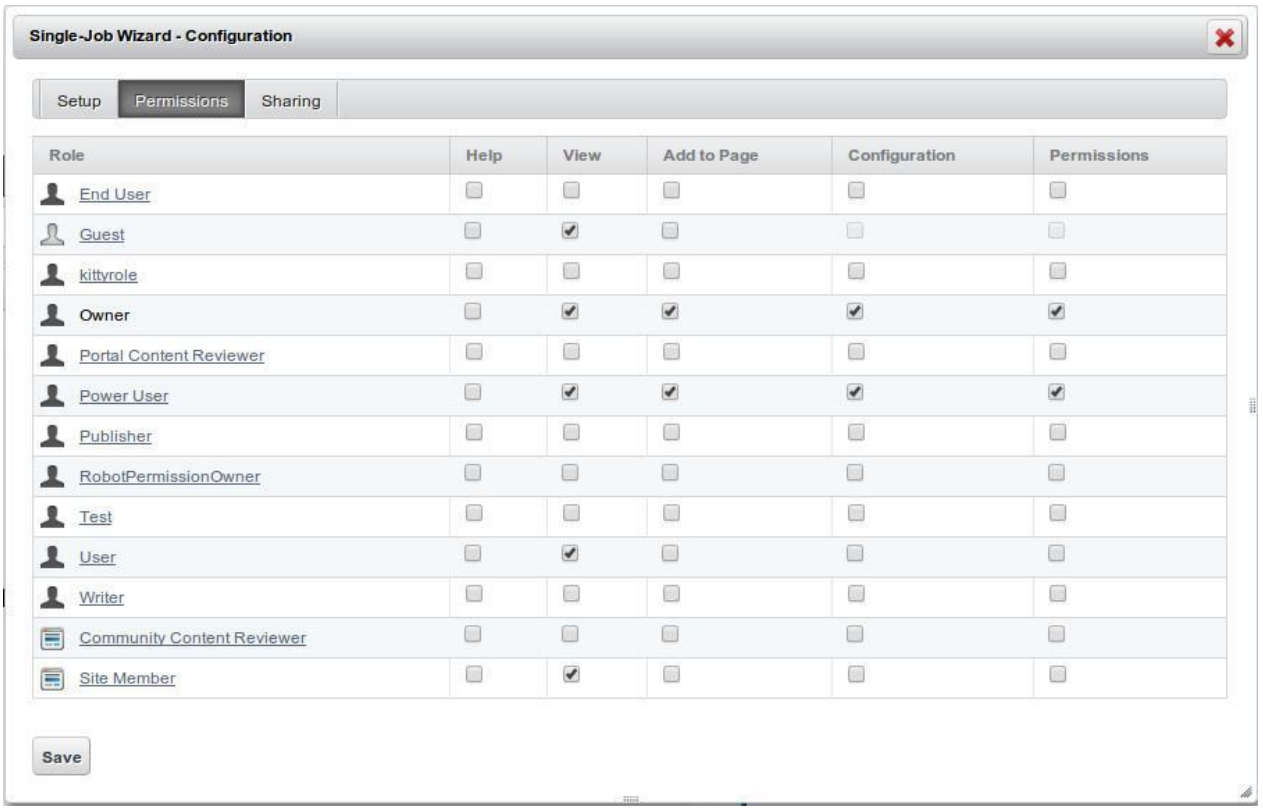


Figure IX.3 Single Job Wizard configuration 3.

Note: More details you find at ASM tab of gUSE SourceForge site: https://sourceforge.net/p/guse/asmsp/wiki/Single%20Job%20Wizard/

Additional Enhancements

Some important and useful solutions are described briefly in this section.

Adding SHIWA Repository

You can use the WS-PGRADE for workflow export/import to/from the SHIWA Repository as of gUSE version 3.5.2. If you need to add a new target SHIWA Repository site into WS-PGRADE, do the followings

1. Sing in to the portal.
2. Navigate to Liferay Portal menu bar (upper horizontal menu bar in the portal window), then select *Go to*/*Control Panel/Plugins Installation/Install More Portlets*/*Upload File* and deploy the WAR file: **wspgrade.war** located in **guse-<versionnumber>.tgz** *webapplication/* subdirectory (Fig. A.1).

The **wspgrade.war** archive file contains among others the **shiwaRepos.xml** where you can set and add SHIWA repositories. The next extract of **shiwaRepos.xml** shows the syntax of *Repos* element in which you can directly specify SHIWA repositories.

```
<Repos>

<Repo name="REPOSITORY1_NAME"
uri="_REPOSITORY1_URL_:_PORT_/_REPOSITORY1_SERVLET_"/>

<Repo name="REPOSITORY2_NAME"
uri="_REPOSITORY2_URL_:_PORT_/_REPOSITORY2_SERVLET_"/>

…

</Repos>
```

If you have got valid credentials to this repository you can export or import workflows to or from the SHIWA Repository.

(The details of SHIWA-specific import and export in WS-PGRADE are described in chapter 7 and 11 of Menu-Oriented Online Help within **Portal User Manual**)

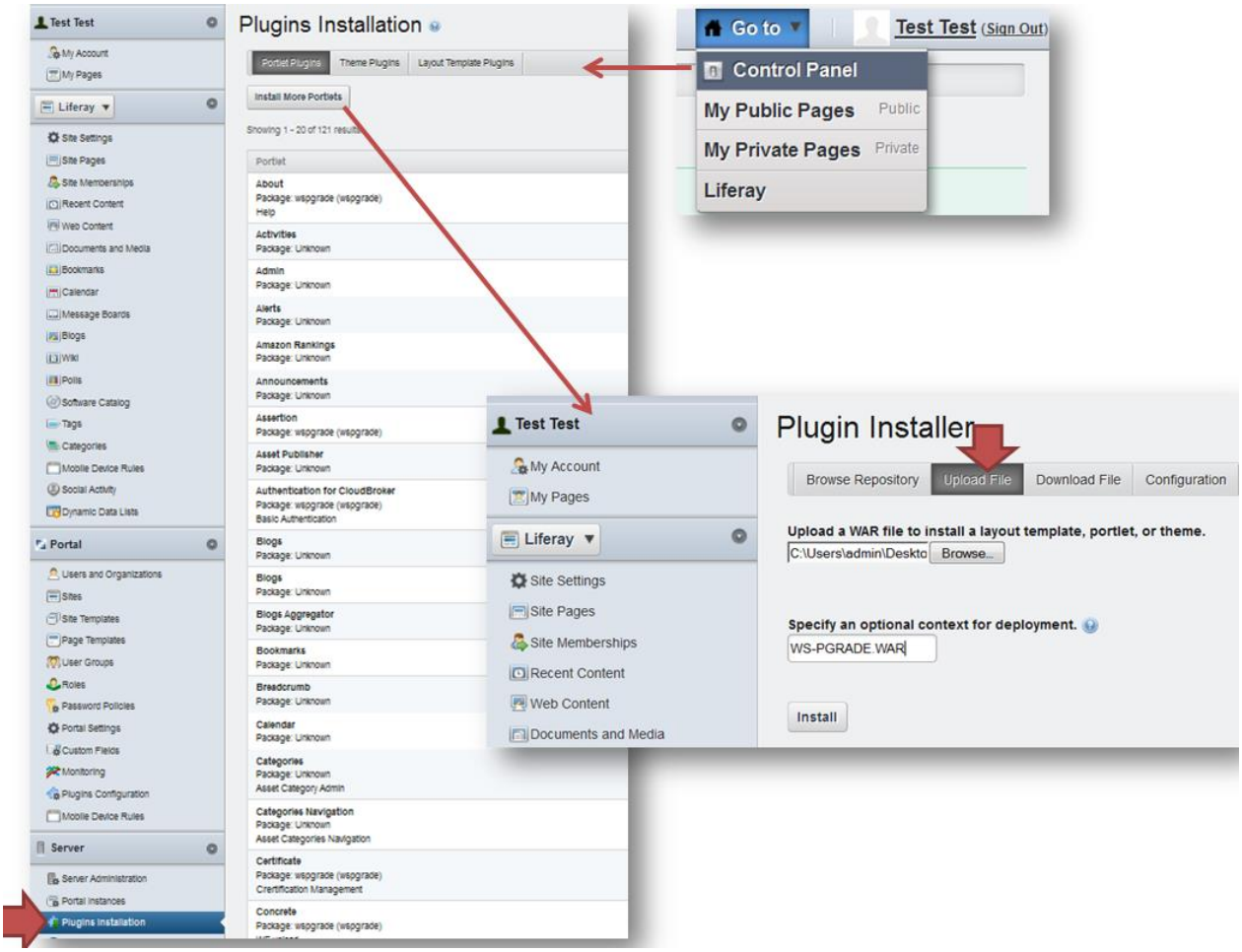


Figure A.1 Using of Plugin Installer function in Liferay Portal to upload the WS-PGRADE WAR file

Site Page (Menu Point) Selection to WS-PGRADE in Liferay

You can select WS-PGRADE-based site pages by a Liferay Portal function. If you don't want to use one or more site pages (in other words: menu points or functions) in your WS-PGRADE portal, you can simple take off that page from visible menu list in the Liferay/WS-PGRADE portal interface. If you want to add that pages later to your other menus, you can simple do it, as well.

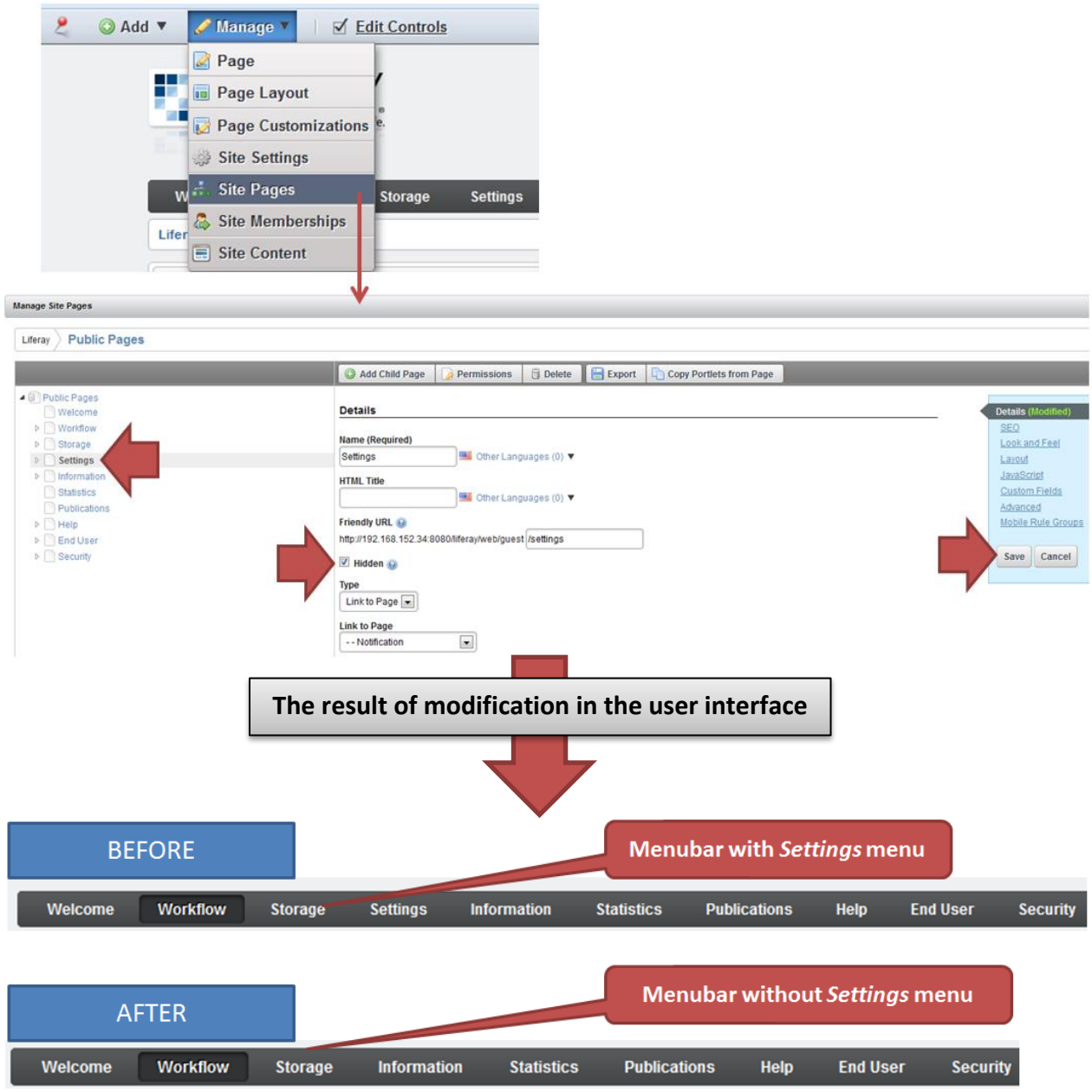


Figure A.2 Hiding the *Settings* menu from menu bar

To execute this Liferay-based administration process, you can use the Liferay menu bar on the top of your WS-PGRADE portal window. Select the *Manage/Site Pages* menu. The *Manage Site Pages* window will appear. Select a wanted main menu or submenu from the left side. Switch on the *Hidden* check box in the middle side of the window if you want to hide the menu (and therefore the related functionality) in the user interface. Switch off the *Hidden* check box if you want to set back the visible state (and therefore the related functionality) of the menu. Save the modified setting by *Save* button in the right side. (Fig. A.2 demonstrates a sample menu hiding process for the *Settings* menu.)

The Text Editor

The system administrator may change displayed texts of WS-PGRADE pages portlets on such places where the JSP responsible for the layout of a portlet containing certain keys. The keys will be substituted by the associated text value. The key-value pairs are stored in the central database of the gUSE system, and the key-value record is maintained by the **Text editor** (in *Settings/Text editor* menu). If there is no matching database item for a given key then the key string will be rendered by the JSP.

Notes:
New key-value items can be defined by selecting the *Add* radio button, filling the three input fields and confirming the operation by clicking the *Submit Query* button where the value entered for Key: should match the key in the JSP; the Descriptor value is a free text entered in order to help the user to find the given item, and the unlabeled text area should contain the defined value text which will substitute the key during the rendering of the proper portlet.

An existing key-value item can be found (and its value eventually edited) selecting the *Find* radio button and using associated check list button. The items in this list are displayed such a way that the key in parentheses follows the descriptor. However the items in the list are enumerated in the lexicographical order of the key part.

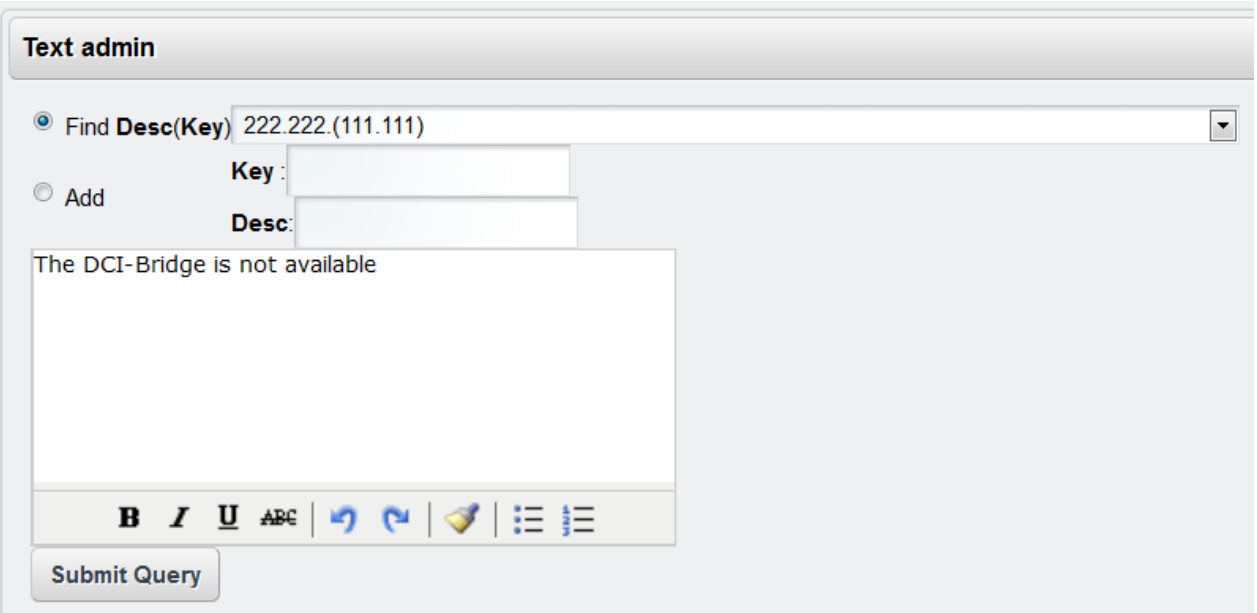


Figure A.3 The Text editor function in WS-PGRADE

Setting System to Local Submitter

For test purposes you can set DCI as the local resource in the following way:

1. Go to **Information/Resources** tab in WS-PGRADE and click on the head icon on the right side. The Tomcat user and password of the back end host will be requested (default is “admin”/”admin”) and DCI Bridge becomes manageable.

2. Select the **Local** tab and in the **Middleware settings** submenu set the **Enable plugin** to “enable” and confirm it by the “Floppy disk” icon. (To realize the middleware changes in settings sign out from WS-PGRADE portal then sign in again. Therefore the portal cache will be updated.)

3. After a successful testing you can change your configuration settings (aka the setting of another DCI Bridge plugin) to target infrastructure by using of **Edit** and **Middleware settings** functions of a corresponding middleware in DCI Bridge configuration interface. (Note: there are two important prerequisites to access resources in case of some middlewares: corresponding preinstalled program - e.g. deployed UIF machine at gLite - and the setting of resource access information in DCI Bridge.)

Notes: to the proper configuration changes you need to know the following general considerations:

You can increase or change the set of supported DCI-s after portal installation by allowing various DCI Bridge plugins or by installing additional DCI Bridge services.

To the proper configuration of a DCI Bridge plugin you can choose an installation location (machine) for DCI Bridge that corresponds to the target DCI infrastructure (e.g. for gLite infrastructure you need an EMI-UI machine).

If you can’t deploy such a machine it remained two possibilities:

1. You need to install the whole portal with the desired settings or

2. You need to take into DCI Bridge configuration process an additional slave DCI Bridge machine where the given plugin is permitted. Therefore, in the legacy master machine you have to set DCI Bridge plugin for sending workflow jobs to slave DCI Bridge machine.

About installation of gUSE/WS-PGRADE see *Installation Wizard Manual*.

Robot Permission Related Logging of Job Submissions

About Robot Permission and Related Logs

The aim in **robot permission** is to perform automated tasks on grids/clouds on behalf of users. Basically, this permission form can be used to identify a person responsible for an unattended service or process acting as client and/or server.

Instead of identifying users, the robot permission identifies trusted applications that can run by workflows from WS-PGRADE. WS-PGRADE supports robot permission for every resource type that is accessible from portal (certainly, the *local* grid type is not require such a permission).

The user who has robot permission is the so called robot permission owner. The robot permission owner has rights to add robot permission association to jobs (about using robot permission in WS-PGRADE see chapter 19 in Menu-Oriented Online Help of Portal User Manual and the Tips & Tricks within WS-PGRADE Cookbook) The entitlement adding to robot permission owner is the same process as the end user role adding (see chapter IV.).

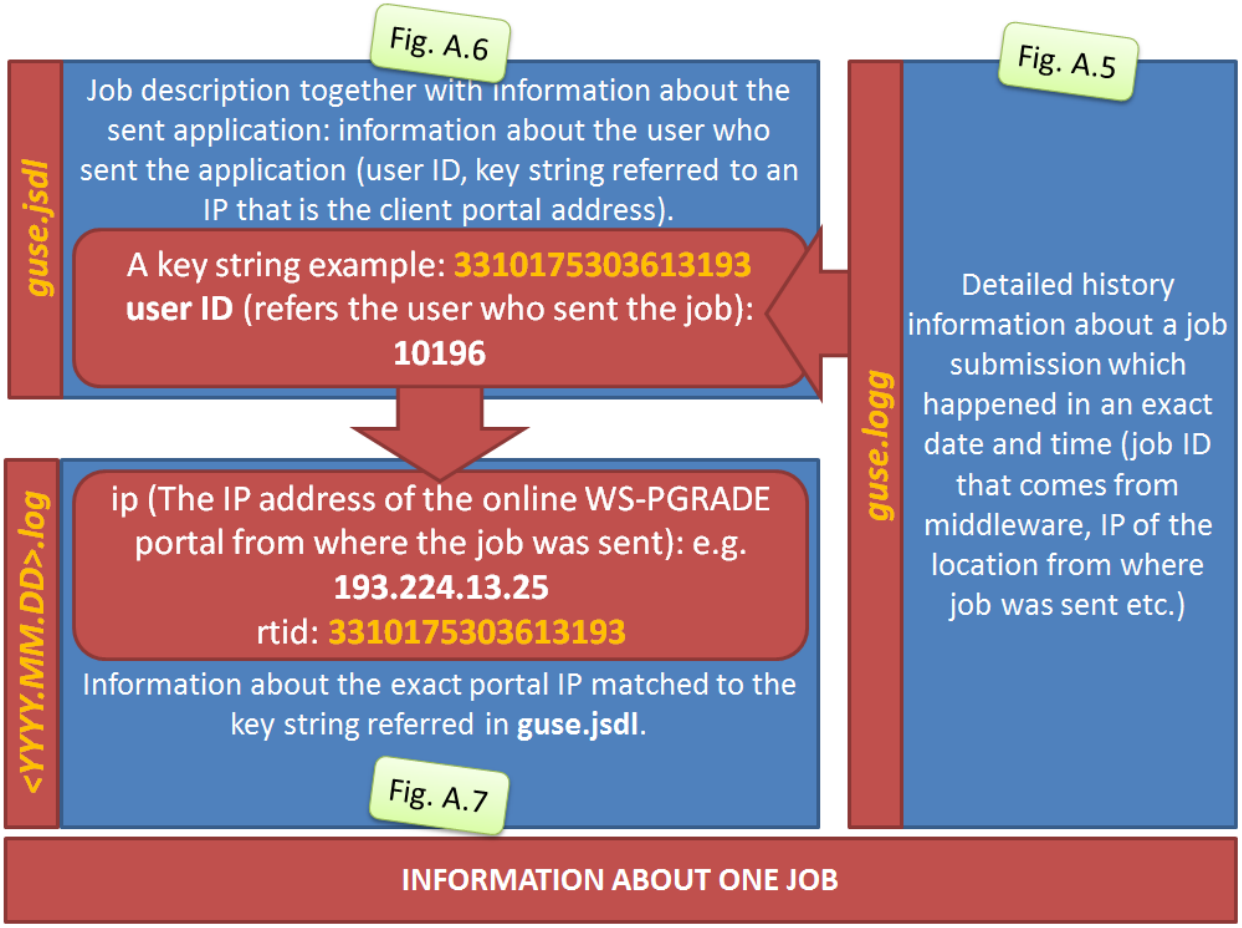


Fig. A.6
guse.jsdl
Job description together with information about the sent application: information about the user who sent the application (user ID, key string referred to an IP that is the client portal address).
A key string example: 3310175303613193
user ID (refers the user who sent the job): 10196
Fig. A.5
guse.logg
Detailed history information about a job submission which happened in an exact date and time (job ID that comes from middleware, IP of the location from where job was sent etc.)
<YYYY.MM.DD>.log
ip (The IP address of the online WS-PGRADE portal from where the job was sent): e.g. 193.224.13.25
rtid: 3310175303613193
Information about the exact portal IP matched to the key string referred in guse.jsdl.
Fig. A.7
INFORMATION ABOUT ONE JOB

Figure A.4 Information retrieval about a robot job from log files

The robot permission related log information is stored for every finished or failed robot job in the DCI Bridge machine within the **tomcat/temp/dci_bridge/robotcertlog** directory, separated by days in **"Year/Month/Day/<x>"** subdirectories where **<x>** is a little integer number. The date refers to the job termination. (Note: As the gUSE/WS-PGRADE system may be distributed on several server instead one more than one Tomcat servers may be involved. Therefore, in the description below the Tomcat controlling the DCI Bridge web application is distinguished from the Tomcat controlling the WFI web application.)

The logs for every job are compressed in ZIP files. The filename is a DCI Bridge-based ID.

The ZIP (Fig. A.6) *may* contain three standard job log files (if the remote resource where the job runs delivers them):

- **gridnfo.log**
- **stdout.log**
- **stderr.log**

The ZIP *must* contain the

- **guse.jsdl** file, which was submitted to DCI Bridge and the
- **guse.logg** file, which contains the events of the job.

Retrieving Information from Log Files

If you want to know who has submitted a robot job to a VO, do the following (the general overview of information retrieval is shown on Fig. A.4 – the detailed explanations are shown on Fig. A.5-A.7):

1. Browse the **guse.logg** files (Fig. A.5) containing the matching job ID (for example **https://grid150.kfki.hu:9000/gBCGIlmsmaE40XTtXfVFBw**)

   The user relevant information in the found **guse.logg** are grouped in the **info** parts of the following items: **logg.job.clientip** = the IP of client, which has submitted the job. It is the IP of the **WFI** server in the case when the job has been sent by gUSE. It is the expected normal case if the job is part of a workflow submitted from the portal or elaborated through the Remote API call interface. However, it may be the IP a foreign host, which is able to communicate with the DCI Bridge directly.

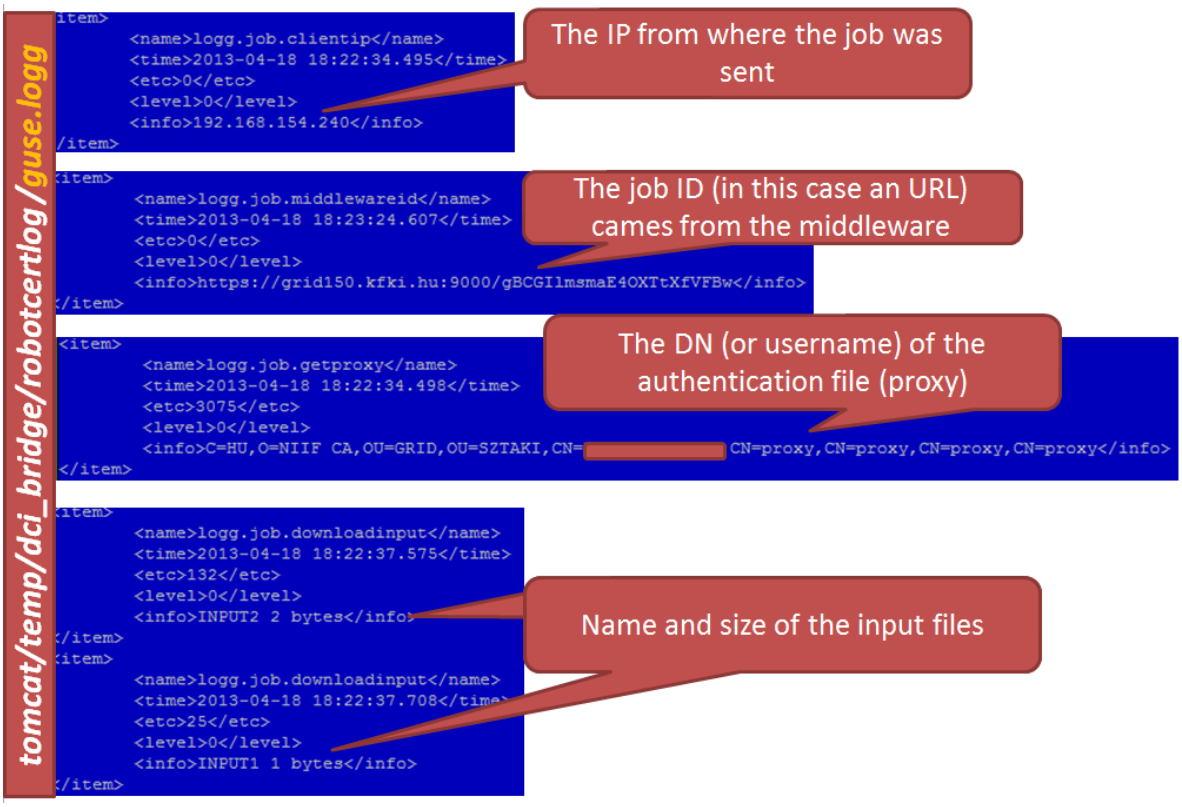


Figure A.5 Some relevant information about a job in guse.logg

2. The "real" sender of the job can be concluded the following way: The sibling job **guse.jsdl** contains all information about the job: the tag **JobDefinition/JobDescription/ Application/ns2:username** contains the Liferay based ID of the end user.

   The IP of the client machine of end user can be identified by the help of the tag **JobDefinition/JobDescription/Application/ApplicationName**: the value of this tag contains a key string containing integer numbers postfixed by the string **zentest**

   An example (Fig. A.6):
   value: **//345491687570196zentest/0/cell/3** -> key string: **345491687570196zentest**

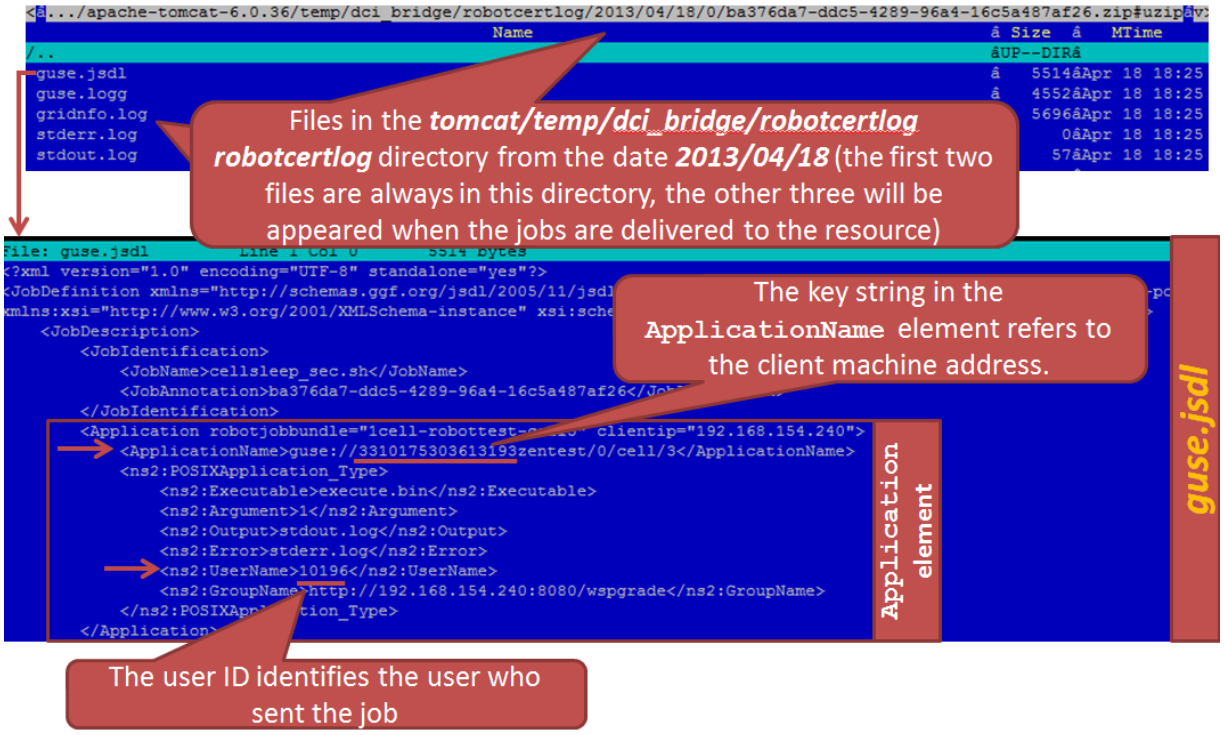

Files in the tomcat/temp/dci_bridge/robotcertlog robotcertlog directory from the date 2013/04/18 (the first two files are always in this directory, the other three will be appeared when the jobs are delivered to the resource)
guse.jsdl
guse.logg
gridnfo.log
stderr.log
stdout.log
The key string in the ApplicationName element refers to the client machine address.
Application element
guse.jsdl
The user ID identifies the user who sent the job

Figure A.6 Robot job related information retrieval from guse.jsdl

This key string must be browsed in the daily LOG files of the WS-PGRADE-machine subdirectory **tomcat/temp/submittedwflog** (see Fig. A.7), which contains a single line entry for each workflows submitted by the gUSE. The matching line contains the requested IP address of the end user have initiated the workflow (and the associated job) submission.

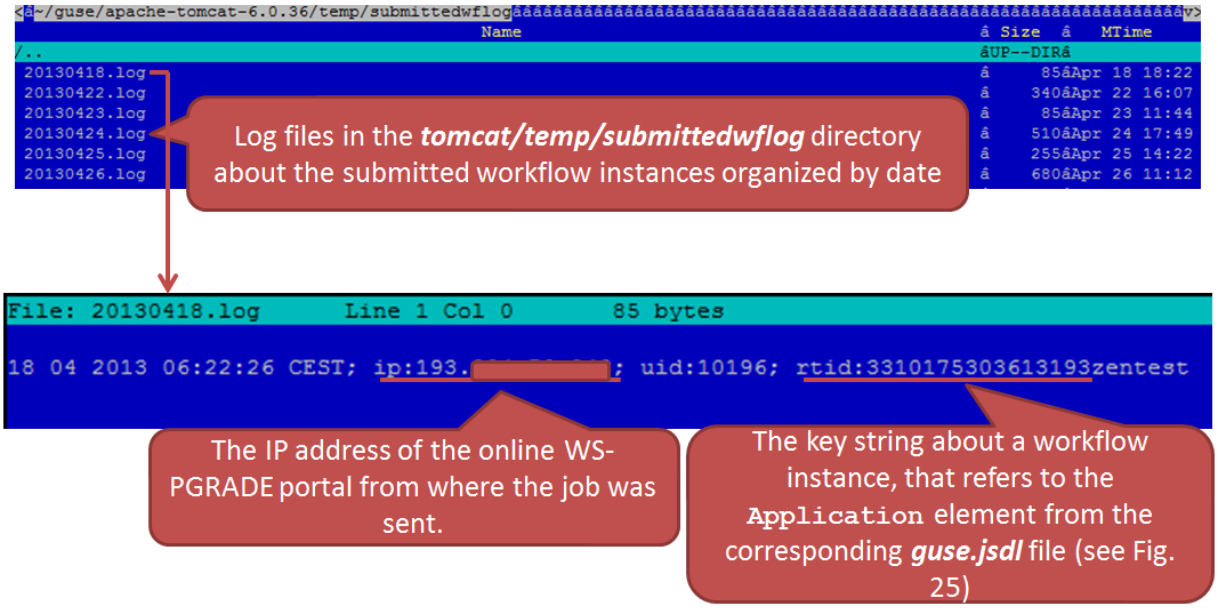

20130418.log
20130422.log
20130423.log
20130424.log
20130425.log
20130426.log
Log files in the tomcat/temp/submittedwflog directory about the submitted workflow instances organized by date
File: 20130418.log Line 1 Col 0 85 bytes
18 04 2013 06:22:26 CEST; ip:193. ; uid:10196; rtid:3310175303613193zentest
The IP address of the online WS-PGRADE portal from where the job was sent.
The key string about a workflow instance, that refers to the Application element from the corresponding guse.jsdl file (see Fig. 25)

Figure A.7 Robot job related information retrieval from log file

Other items of the matched **guse.logg** file may be of interest (see Fig. A.5):

- **logg.job.getproxy** = Hint of the authorization of the job: the DN (or username) of the authentication file (proxy)
- **logg.job.downloadinput** = Name and size of the input file(s)

Adding and Removing User Roles

The administration of user roles of gUSE/WS-PGRADE is at Liferay portal administration level (About the end user role administration see chapter IV.). There are three general user roles in gUSE (certainly, it is possible to define other user roles by set various permissions, but these three main roles can capture the essential user roles in WS-PGRADE):

- **End user** role (fixed name for role setting: *End User*): The end user needs only a restricted manipulation possibility. He/she wants to get the application "off the shelf", to trim it and submit it with minimal effort.
- **Workflow developer user** or **(full) power user** role (fixed name: *Power User*): This user builds and tailors the application to be as comfortable as possible for the end user.
- **Robot permission owner** role (fixed name: *RobotPermissionOwner*)**:** The robot permission owner is a special trusted user for creation the so called robot permission association for a trusted application.

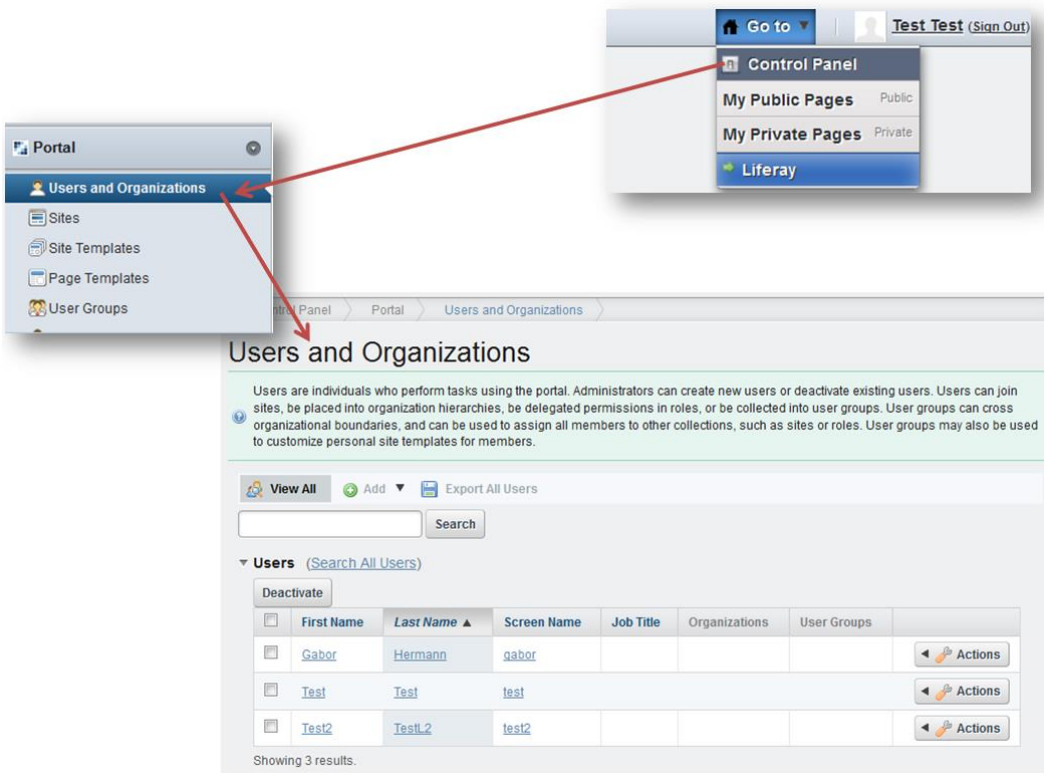


Figure A.8 Adding a user role I.

If you want to add or remove a role as administrator for a user follow the next steps (Fig. A.8-A.10.):

1. Choose the *Go to/Control Panel* menu on the Liferay portal administration interface on the top of a portal window.
2. Click on *Users and Organizations* menu on the left side. In the appearing window choose a user by clicking on her/his name then choose the *Roles* menu within the *User Information* menu group in the right side of the next window.
3. Choose the *Select* link in the end of *Regular Roles* list.
4. Choose a corresponding predefined user role from the appearing list.
5. For removing a role use the *Remove* function at the selected user role row of the *Regular Roles* list (Fig. A.10).
6. Save the changes by clicking on the *Save* button on the upper right side.

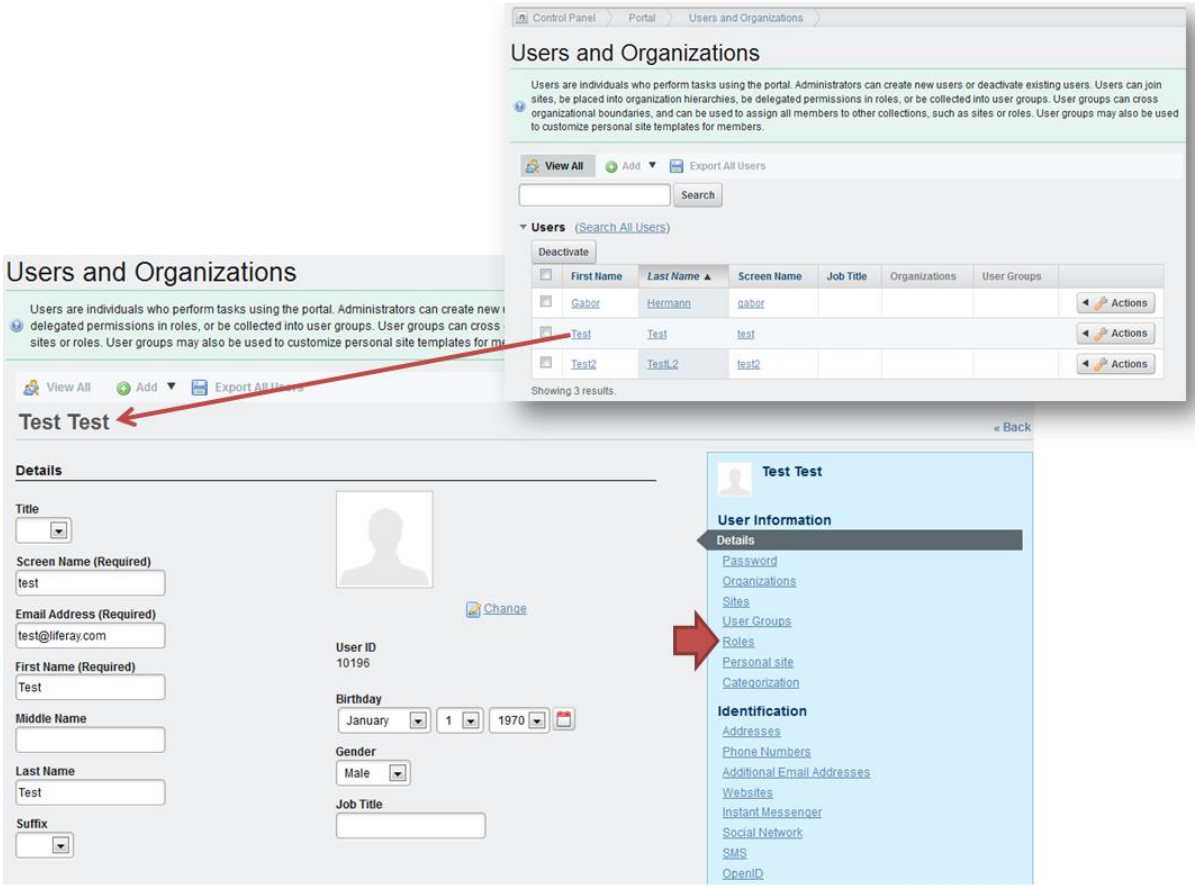


Figure A.9 Adding a user role II.

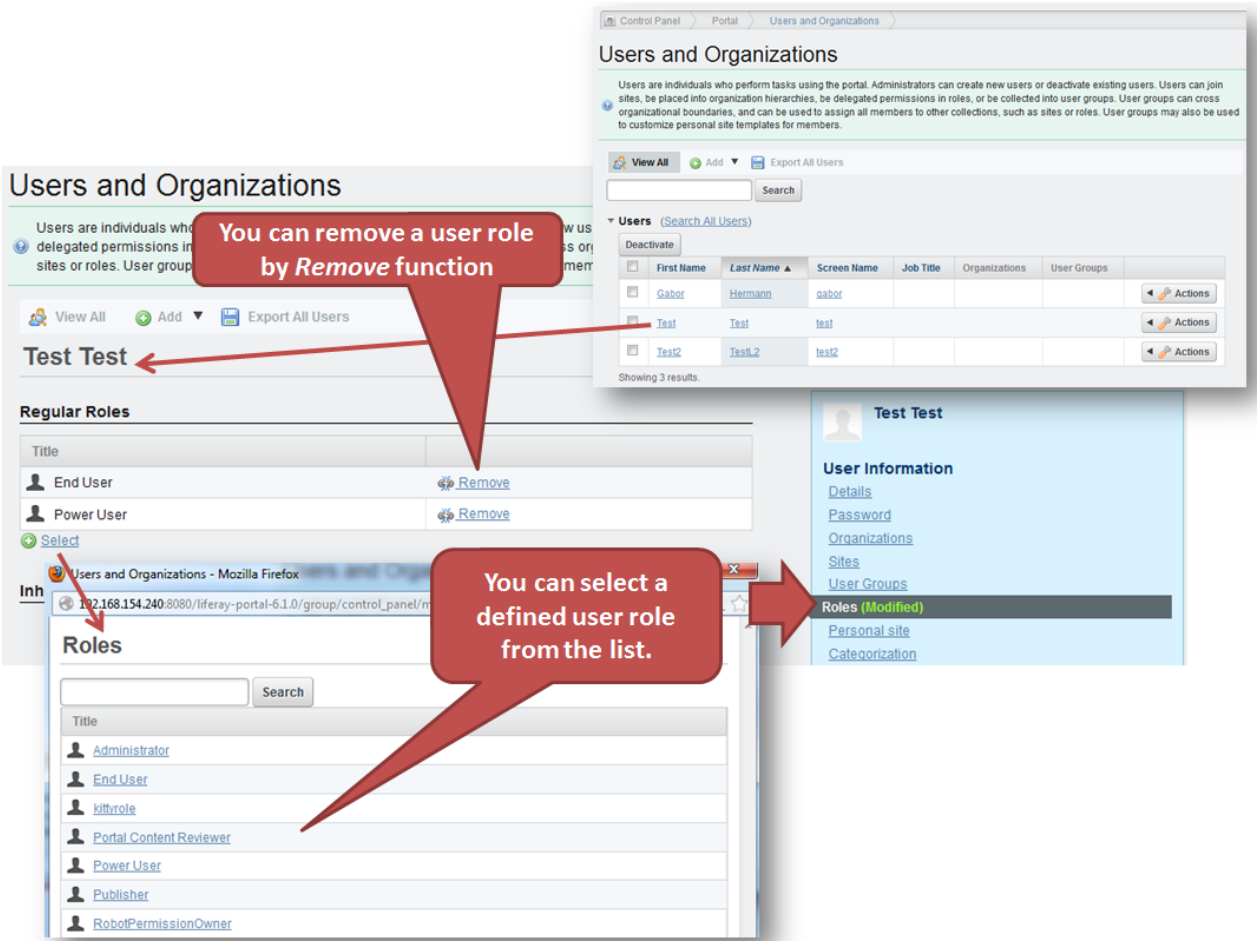


Figure A.10 Adding a user role to a user III.

Supported Protocols for Using of Remote Executable and Input

During the job configuration process you need to define your executable code of binary within the *Concrete/Job Executable* tab. You can do it not only by uploading binary from your local machine (*Local* option) but by adding the corresponding remote path of the executable (*Remote* option).

You can use the *Remote* option to the following Grid *Types* together with the next protocols for the *Executable code of binary* definition:

Grid Type	Supported Protocols
gLite	file, http, lfn
GT2	file, http, gsiftp
GT4	file, http, gsiftp
GT5	file, http, gsiftp
UNICORE	http, idbtool
ARC	http
LSF	http
PBS	file, http
Local	file, http

You can also use the *Remote* option (by selecting the "*Remote*" icon) to input port definition within the *Concrete/Job I/O* tab (Figure 14.2). In this case you can use the following Grid Types and protocols:

Grid Type	Supported Protocols
gLite	http, lfn
GT2	http, gsiftp
GT4	http, gsiftp
GT5	http, gsiftp
UNICORE	http, xtreemfs
ARC	http
LSF	http
PBS	http
Local	http

New Parameter Definition to a Middleware

You can define new parameters (e.g. JDL or RSL parameters) for each supported middleware type in WS-PGRADE *Text admin* window (*Settings/Message & Label Editor*) in the following way:

For each parameter must be defined two key-value pairs.

- for the help text and
- for the parameter key.

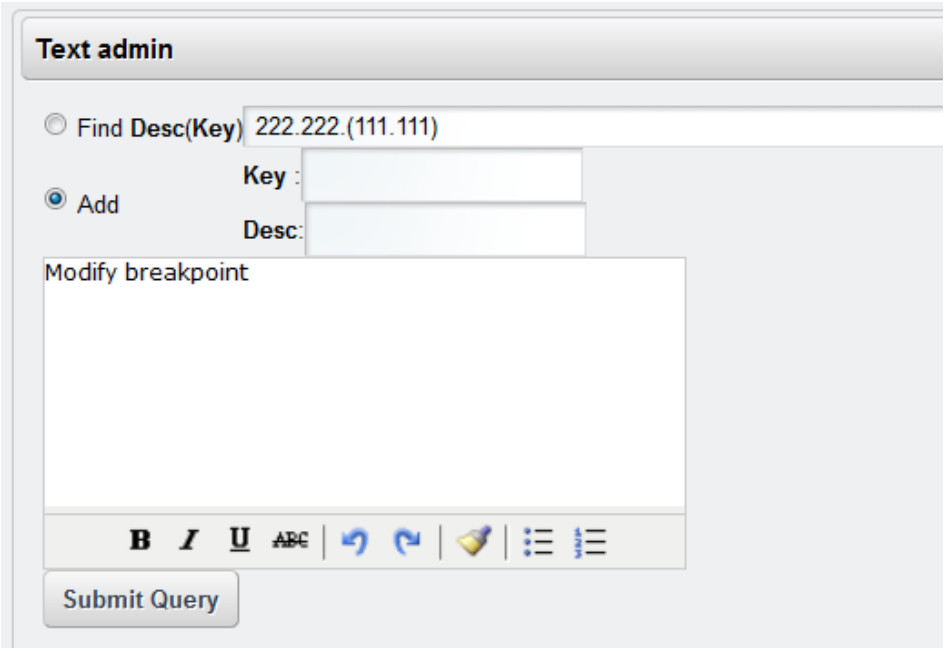


Figure A.11 Parameter adding to a middleware in the Text admin window

See an example in case of **gLite** middleware:

- Add the key *CPUNumber* for gLite middleware: Select the radio button *Add*
- Set the following parameters:
  - **Key**: *glite.textCPUNumber*
  - **Desc**: --
  - **Value** (the content of the text field): *The CPUNumber attribute is an integer greater than 1...*
- Click on *Submit Query*.

The keys will be stored in the database.

If the previous values are disappeared, restart your WS-PGRADE.

Logging with log4j

For better troubleshooting and debugging gUSE uses the **log4j**-based solution.

The **Apache log4j** is a logging library for Java. With log4j it is possible to enable logging at runtime without modifying the application binary. The log4j package is designed so that these statements can remain in shipped code without incurring a heavy performance cost. Logging behavior can be controlled by editing a configuration file, without touching the application binary.

The practical advantage of log4j-logging is to set the level of logging. Therefore you can avoid the logging of unnecessary events.

Note: Technically, the target file of logging is not changed with the introducing of log4j: the **`catalina.out`** file.

In gUSE you can use the **INFO**, **ERROR**, and **DEBUG** properties to set the desired logging level to the given component classes.

Example: To set a gUSE component for detailed debugging, do the followings:

1. Go to the corresponding gUSE component (to the **`<Tomcat_Home>/webapps/<Component_Name>/WEB-INF/classes`** directory)
2. Open the **log4j.properties** file
3. Modify the next row:

   ```
   log4j.category.hu.sztaki=INFO
   ```

   to this:

   ```
   log4j.category.hu.sztaki=DEBUG
   ```

4. Restart Tomcat.

Note: log4j provides many other configuration settings. About this you find details here: https://logging.apache.org/log4j/1.2/manual.html

Administrative Tasks for Job Submission to SZTAKI Desktop Grid

In order to submit jobs to **SZTAKI Desktop Grid** (e.g. **SZTAKI DG** - http://szdg.lpds.sztaki.hu/szdg/), you have to:

- first deploy the necessary applications on the desktop grid (it is not necessary to deploy master components for them, as **3G Bridge (Generic Grid-Grid Bridge**) will act as a master here.
- add the applications to 3G Bridge's **cg_algqueue** table (see here: http://doc.desktopgrid.hu/doku.php?id=manual:gbac#server_side3g_bridge_configuration)
- configure the 3G Bridge's accessibility as a new **BOINC** resource in the DCI Bridge. (See *chapter 2.9* in the *DCI Bridge Manual*.)

This will make sure, that:

- 3G Bridge is aware of what applications you have deployed on BOINC
- gUSE/WS-PGRADE will be able to contact the 3G Bridge to query the list of usable applications,
- users of WS-PGRADE will be able to select from this set of applications during their workflows' configuration.

Additional information:

3G Bridge Installation Manual: http://doc.desktopgrid.hu/doku.php?id=manual:3gbridge
3G Bridge Manual: http://doc.desktopgrid.hu/doku.php?id=component:3gbridge

Portlet Adding for WS-PGRADE Portal

When you need to define a new menu into your WS-PGRADE portal, you technically need to add a portlet to others. You have to perform this simple operation by Liferay tools (Chapter III. describes the difference between Liferay- and WS-PGRADE-based settings).

The next step sequence represents a menu adding process in case of setting to XSEDE middleware. The sample process also describes the function adding (in this case the authentication function) to the created menu:

1. Create a new menu item: go to the top of portal window in the Liferay menu bar and click on **Manage/Site Pages**. Select **Security** from the menus in the left side and then click **Add Child Page**.
2. Specify the name "**XSEDE**" and click **Add Page**. Click **Save** on the right side. The new menu will appear within the **Security** menu.
3. Select **Security/XSEDE** from your portal, click **Add/More..** from the Liferay portal menu in the top and select **SZTAKI.wspgrade portlets** from the list. From the appearing sublist select **Authentication for XSEDE** and click **Add**.
4. Click **Manage/Page Layout** in the top of window and select the **1 Column** mode from the layouts. Finally, click **Save** on the right side.